# Kooperative Angriffserkennung in drahtlosen Ad-hoc- und Infrastrukturnetzen

## Anforderungsanalyse, Systementwurf und Umsetzung

**Dissertation**

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

**Dipl.-Inform. Stephan Groß**
geboren am 25. April 1973 in Saarburg

**Gutachter:**

| | |
|---|---|
| Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill | Technische Universität Dresden |
| Prof. Dr. rer. nat. Andreas Pfitzmann | Technische Universität Dresden |
| Prof. Dr.-Ing. Ahmad-Reza Sadeghi | Ruhr-Universität Bochum |

Tag der Einreichung: 22. September 2008
Tag der Disputation: 1. Dezember 2008

Dresden, im April 2009

# Zusammenfassung

Mit der zunehmenden Verbreitung mobiler Endgeräte und Dienste ergeben sich auch neue Herausforderungen für ihre Sicherheit. Diese lassen sich nur teilweise mit herkömmlichen Sicherheitsparadigmen und -mechanismen meistern. Die Gründe hierfür sind in den veränderten Voraussetzungen durch die inhärenten Eigenschaften mobiler Systeme zu suchen. Die vorliegende Arbeit thematisiert am Beispiel von Wireless LANs die Entwicklung von Sicherheitsmechanismen für drahtlose Ad-hoc- und Infrastrukturnetze. Sie stellt dabei den umfassenden Schutz der einzelnen Endgeräte in den Vordergrund, die zur Kompensation fehlender infrastruktureller Sicherheitsmaßnahmen miteinander kooperieren.

Den Ausgangspunkt der Arbeit bildet eine Analyse der Charakteristika mobiler Umgebungen, um grundlegende Anforderungen an eine Sicherheitslösung zu identifizieren. Anhand dieser werden existierende Lösungen bewertet und miteinander verglichen. Der so gewonnene Einblick in die Vor- und Nachteile präventiver, reaktiver und angriffstoleranter Mechanismen führt zu der Konzeption einer hybriden universellen Rahmenarchitektur zur Integration beliebiger Sicherheitsmechanismen in einem kooperativen Verbund. Die Validierung des Systementwurfs erfolgt anhand einer zweigeteilten prototypischen Implementierung.

Den ersten Teil bildet die Realisierung eines verteilten Network Intrusion Detection Systems als Beispiel für einen Sicherheitsmechanismus. Hierzu wird eine Methodik beschrieben, um anomalie- und missbrauchserkennende Strategien auf beliebige Netzprotokolle anzuwenden. Die Machbarkeit des geschilderten Ansatzes wird am Beispiel von infrastrukturellem WLAN nach IEEE 802.11 demonstriert.

Den zweiten Teil der Validierung bildet der Prototyp einer Kooperations-Middleware auf Basis von Peer-to-Peer-Technologien für die gemeinsame Angriffserkennung lose gekoppelter Endgeräte. Dieser kompensiert bisher fehlende Mechanismen zur optimierten Abbildung des Overlay-Netzes auf die physische Struktur drahtloser Netze, indem er nachträglich die räumliche Position mobiler Knoten in die Auswahl eines Kooperationspartners einbezieht. Die zusätzlich definierte Schnittstelle zu einem Vertrauensmanagementsystem ermöglicht die Etablierung von Vertrauensbeziehungen auf Kooperationsebene als wichtige Voraussetzung für den Einsatz in realen Umgebungen. Als Beispiel für ein Vertrauensmanagementsystem wird der Einsatz von Reputationssystemen zur Bewertung der Verlässlichkeit eines mobilen Knotens diskutiert. Neben einem kurzen Abriss zum Stand der Forschung in diesem Gebiet werden dazu zwei Vorschläge für die Gestaltung eines solchen Systems für mobile Ad-hoc-Netze gemacht.

# Abstract

The increasing deployment of mobile devices and accompanying services leads to new security challenges. Due to the changed premises caused by particular features of mobile systems, these obstacles cannot be solved solely by traditional security paradigms and mechanisms. Drawing on the example of wireless LANs, this thesis examines the development of security mechanisms for wireless ad hoc and infrastructural networks. It places special emphasis on the comprehensive protection of each single device as well as compensating missing infrastructural security means by cooperation.

As a starting point this thesis analyses the characteristics of mobile environments to identify basic requirements for a security solution. Based on these requirements existing preventive, reactive and intrusion tolerant approaches are evaluated. This leads to the conception of a hybrid and universal framework to integrate arbitrary security mechanisms within cooperative formations. The resulting system design is then validated by a twofold prototype implementation.

The first part consists of a distributed network intrusion detection system as an example for a security mechanism. After describing a methodology for applying anomaly- as well as misuse-based detection strategies to arbitrary network protocols, the feasibility of this approach is demonstrated for IEEE 802.11 infrastructural wireless LAN.

The second part of the validation is represented by the prototype of a P2P-based cooperation middleware for collaborative intrusion detection by loosely coupled devices. Missing mechanisms for the improved mapping of overlay and physical network structures are compensated by subsequently considering the spatial position of a mobile node when choosing a cooperation partner. Furthermore, an additional interface to an external trust management system enables the establishment of trust relationships as a prerequisite for a deployment in real world scenarios. Reputation systems serve as an example of such a trust management system that can be used to estimate the reliability of a mobile node. After outlining the state of the art, two design patterns of a reputation system for mobile ad hoc networks are presented.

# Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Professur Rechnernetze der Technischen Universität Dresden. Zu ihrem Gelingen haben eine Vielzahl von Personen auf unterschiedlichste Weise beigetragen, denen ich an dieser Stelle herzlich danken möchte.

An erster Stelle danke ich Prof. Alexander Schill für seine Unterstützung und das mir entgegengebrachte Vertrauen während der letzten Jahre. So war etwa die Vereinbarkeit von Beruf und Familie dank flexibler Arbeitszeiten an seinem Lehrstuhl bereits selbstverständlich, bevor sie auf der politischen Agenda stand. Sehr geholfen haben mir auch sein beharrliches Drängen auf den Fortgang meiner Arbeit und die schnelle Kommentierung von Manuskriptentwürfen.

Prof. Andreas Pfitzmann hat durch sein herzliches Willkommen bei meinem ersten Besuch in Dresden wesentlichen Anteil daran, dass dieses Dissertationsvorhaben überhaupt zustande gekommen ist. Darüber hinaus danke ich ihm für zahlreiche hilfreiche Kommentare und die freundliche Begleitung bei der Fertigstellung dieser Arbeit. Prof. Ahmad-Reza Sadeghi danke ich für seine Bereitschaft, das Koreferat für diese Arbeit zu übernehmen.

Meinen (ehemaligen) Kollegen am Lehrstuhl Rechnernetze möchte ich für das ausgesprochen freundschaftliche und konstruktive Arbeitsumfeld danken. Stellvertretend seien hier René Neumerkel und Sandro Reichert genannt, die bereits als Diplomanden großen Anteil an dieser Arbeit hatten. Im Rahmen zahlreicher Diskussionen haben sie immer wieder meinen Blick auf mögliche Probleme bei der Umsetzung meines Ansatzes in die Praxis geschärft. Auch mein Zimmergenosse Marius Feldmann war jederzeit bereit, meine Arbeit zu hinterfragen. Außerdem hat er das vorliegende Manuskript einem sorgfältigen Lektorat unterzogen.

Vor allem aber möchte ich mich bei meiner Familie bedanken, allen voran bei meiner Frau Sandra, die fachlich mitdiskutiert und mich auch sonst in jeder erdenklichen Weise unterstützt hat, sowie unseren beiden Kindern Jonathan und Finnya. Ihr Verständnis, wenn Papa wieder an die Arbeit musste, aber auch ihr Beharren auf Familienzeit haben mir gerade in schwierigen Phasen sehr geholfen. Darüber hinaus danke ich meinen Eltern und allen anderen Familienmitgliedern für ihren Ansporn und ihr Verständnis ob meiner häufig knapp bemessenen Zeit. Mein Bruder Andreas war außerdem so nett, weite Teile meines Manuskripts Korrektur zu lesen.

Dresden, im September 2008 Stephan Groß

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# 1 Einleitung und Motivation

In den vergangenen Jahren hat die Verbreitung mobiler Endgeräte rasant zugenommen. Dank steigender Leistungsfähigkeit werden immer mehr sensitive Informationen auf Laptops, PDAs und Smartphones gespeichert und verarbeitet. Auf der anderen Seite werden verstärkt drahtlose Netze eingesetzt, da sie flexibler und kostengünstiger zu installieren sind als kabelgebundene. Hierbei werden jedoch häufig die im Vergleich zu herkömmlichen kabelbasierten Netzen geänderten Voraussetzungen hinsichtlich der Sicherheit solcher Systeme vernachlässigt. Das Scheitern des IEEE 802.11 WEP-Standards belegt dies eindringlich [BHL06, TWP07]. Zwar wurden dessen Unzulänglichkeiten mit der Einführung von Wi-Fi Protected Access (WPA) und IEEE 802.11i weitestgehend behoben, doch zeigen aktuelle Untersuchungen, dass sich die neuen Methoden nur langsam durchsetzen [Dö06]. Die Gründe hierfür sind vielfältig und reichen von der fehlenden Unterstützung der neuen Standards durch bereits vorhandene Systeme über Schwierigkeiten im Umgang mit den neuen Sicherheitstechniken bis hin zum völlig fehlenden Bewusstsein für die Risiken beim Umgang mit drahtlosen Netzen.

Aber selbst wenn präventive Sicherheitstechniken eingesetzt werden, bleiben Risiken bestehen. So können mobile Endgeräte, nomen est omen, von ihrem Besitzer an beliebigen Orten eingesetzt werden. Die Sicherheitsgrenzen werden damit durchlässiger, da mobile Endgeräte nicht immer in einem geschützten Umfeld betrieben werden. Darüber hinaus sind auch bestehende Netze gefährdet, da mittels mobiler Geräte zentrale Sicherheitsmechanismen wie etwa Firewalls umgangen und somit Angriffe quasi durch die Hintertür ermöglicht werden [ZLH03]. Eine weitere Gefahr für lokale Netze stellen fehlerhaft konfigurierte Geräte dar, die das Routing zwischen der drahtlosen Netzwerkschnittstelle und dem Festnetz ermöglichen, oder der Anschluss von nichtautorisierten

WLAN-Zugangsknoten, sogenannten *Rogue Access Points*. Diese Gefahren können durch präventive Maßnahmen nicht mit letzter Sicherheit ausgeschlossen werden, weshalb für einen umfassenden Schutz die Überwachung der Netzumgebungen erforderlich ist. Hierfür werden in der Regel Intrusion Detection Systeme (IDS) verwendet. Doch auch diese werden durch die veränderten Voraussetzungen aufgrund der zunehmenden Mobilität der eingesetzten Endgeräte vor neue Herausforderungen gestellt, die bisher nur unzureichend berücksichtigt werden.

Das Überdenken bestehender Konzepte und Verfahren zur Sicherung von Rechnernetzen erscheint vor diesem Hintergrund also dringend geboten. Mit der Entwicklung einer kooperativen Architektur zur Erkennung von Angriffen in drahtlosen Ad-hoc- und Infrastrukturnetzen leistet die vorliegende Arbeit einen Beitrag zu dieser Diskussion. Bevor jedoch Zielsetzung und Gliederung der Arbeit weiter konkretisiert werden, sollen die folgenden beiden Abschnitte zunächst das Problemfeld genauer umreißen.

## 1.1 Sicherheit in Rechnernetzen

Allgemein versteht man unter *Informationssicherheit* die Eigenschaft eines Systems, nur solche Systemzustände anzunehmen, die zu keiner unautorisierten Informationsveränderung oder -gewinnung führen [Eck03]. Das Ziel von Informationssicherheit ist also die Vertraulichkeit, Integrität und Verfügbarkeit von Informationen und der sie verarbeitenden Systeme sicherzustellen. Eine *(Sicherheits-)Richtlinie* (engl. *(security) policy*) beschreibt hierzu, welche Handlungen und Ereignisse zulässig sind oder nicht [Bis03]. Die Durchsetzung einer solchen Sicherheitsrichtlinie erfolgt mittels sogenannter *Sicherheitsmechanismen* (engl. *security mechanisms)*, wozu nicht nur technische, sondern auch organisatorische Maßnahmen zählen. Diese können in drei Klassen eingeteilt werden: Prävention, Überwachung und Reaktion.

*Prävention* beinhaltet die Implementierung von Mechanismen, die von vornherein die Verletzung von Schutzzielen verhindern. Beispiele hierfür sind Firewalls, Applikationsfilter oder auch VPN-Gateways. Sind präventive Maßnahmen korrekt implementiert und die ihnen zugrunde liegenden Annahmen gültig, stellen sie einen effektiven und umfassenden Schutz dar. Diese Einschränkung macht deutlich, warum trotzdem überwachende Maßnahmen notwendig sind. Weder die korrekte Implementierung noch die Berücksichtigung aller relevanten Eventualitäten lässt sich heute in der Praxis sicherstellen. Außerdem gibt es Sicherheitsanforderungen, die sich nicht vollständig präventiv durchsetzen lassen. Als Beispiel hierfür sei die Verfügbarkeit verteilter Systeme genannt. Darüber hinaus ermöglicht die *Überwachung* und Erkennung von Schutzzielverletzungen die Einschätzung der Effektivität von präventiven Maßnahmen. Wurde eine Schutzzielverletzung erkannt, erfordert dies in aller Regel eine *Reaktion*. Diese kann von der einfachen Wiederherstellung des letzten korrekten Systemzustands bis hin zu einer direkten Erwiderung reichen. Zusammen bilden diese drei Klassen von Sicherheitsme-

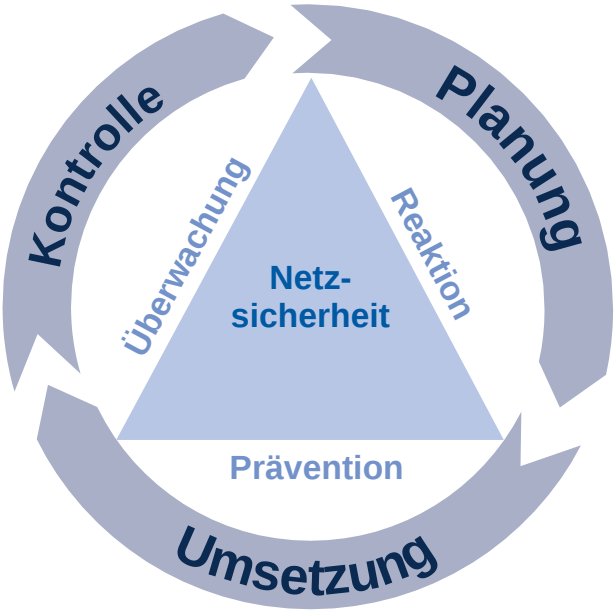



Abbildung 1.1: Die Dreigestalt der Netzsicherheit

chanismen die Grundlage für alle Konzepte zum Schutz von Rechnernetzen [Can01]. Abbildung 1.1 verdeutlicht dies, indem sie einen Bezug zu den dynamischen Aspekten der Informationssicherheit in Anlehnung an das PDCA-Modell nach W. Edwards Deming und Walter Shewhart herstellt [ISO05].[1] Untersucht man die Art und Weise genauer, wie die unterschiedlichen Mechanismen zur Sicherung von Rechnernetzen eingesetzt werden, kann man eine Reihe von Beobachtungen anstellen:

*Schutz des Netzes als Ganzes:* Obwohl Netzwerke per Definition aus einer Vielzahl autonomer Systeme bestehen, werden sie bei der Planung und Realisierung von Netzsicherheitsmaßnahmen als zusammenhängendes System betrachtet und daher auch als Ganzes geschützt.

*Absicherung der Systemgrenzen:* Ein essentielles Prinzip stellt zudem die Absicherung an den Systemgrenzen dar, wobei man sich meist am TCP/IP-Referenzmodell orientiert. So erfolgt der Schutz lokaler Netze beispielsweise durch die Kontrolle der Wegewahl und Paketweiterleitung auf der Vermittlungsschicht, etwa durch den Einsatz von Firewalls. Dies bedeutet aber, dass die Netztopologie in die Konfiguration der Sicherheitsmechanismen zumindest implizit einfließt oder diese

[1] Im Gegensatz zum herkömmlichen viergeteilten Deming-Kreis wird hier nur zwischen den drei Phasen Planung, Umsetzung und Kontrolle unterschieden. Der Schritt Aktion (engl. *Act*) des ursprünglichen Modells wird stattdessen anteilig der Planungs- und Umsetzungsphase zugeordnet.

sogar entsprechend den Sicherheitsanforderungen geplant wird. Die Konzentration der Sicherheitsmaßnahmen auf einzelne Schichten des Referenzmodells erfordert zudem eine sorgfältige Betrachtung der Annahmen über die Sicherheit der darunterliegenden Schichten.

*Einsatz dedizierter Sicherheitssysteme:* Die Durchsetzung der Netzsicherheit erfolgt in aller Regel mittels dedizierter Sicherheitssysteme. Beispiele hierfür sind Firewalls, Applikationsfilter, VPN-Gateways oder Intrusion Detection Systeme. Dies liegt einerseits an dem oftmals signifikanten Ressourcenbedarf von Sicherheitsmechanismen, der einen konkurrierenden Betrieb mit anderen Diensten erschwert oder sogar unmöglich macht. Zum anderen sind die Sicherheitssysteme selbst bevorzugte Angriffsziele und müssen daher besonders geschützt werden.

*Zentrale Kontrolle:* Infolge ihres hierarchischen Aufbaus werden heutige Netze zudem von zentraler Stelle kontrolliert und gewartet. Dies gilt auch für die eingesetzten Sicherheitsmechanismen, deren Konfiguration und Überwachung zudem ein hohes Maß an Expertenwissen voraussetzen.

Ein solches Sicherheitsmodell wird allgemein *Perimeterverteidigung* (engl. *Perimeter Defense*) genannt [Can01]. Nicht zuletzt aus ökonomischen Gründen hat sich dieser Ansatz in der Praxis durchgesetzt, obwohl er einige ernstzunehmende Schwächen aufweist: Zum einen bietet er keinerlei Schutz vor Insider-Angriffen, zum anderen wird mit der Kompromittierung des Perimeterschutzes auch der des übrigen Systems ausgehebelt. Im folgenden Abschnitt werden den angestellten Beobachtungen die besonderen Eigenschaften mobiler Umgebungen gegenüber gestellt, um so die Unzulänglichkeiten einer Perimeterverteidigung für deren Schutz darzulegen. Stattdessen wird eine Reihe von Forderungen für die Realisierung eines robusteren, tiefergehenden Sicherheitsmodells für mobile Umgebungen aufgestellt und so die weitere Zielsetzung motiviert.

## 1.2 Besonderheiten mobiler Umgebungen

Eine hervorstechende Besonderheit mobiler Umgebungen stellt die Unterscheidung zwischen den Betriebsmodi Ad-hoc und Infrastruktur dar. Im *Infrastruktur-Modus* werden mobile Rechner über feste Basisstationen, sogenannte *Access Points (AP)*, mit einer bestehenden Netzinfrastruktur verbunden. Auch die Kommunikation zwischen einzelnen mobilen Endgeräten erfolgt hierbei immer über den Access Point. Der einzige Unterschied zu klassischen kabelbasierten Netzen stellt damit die Funkübertragung zu den Endgeräten dar. Hinsichtlich der in bestehenden Systemen häufig getroffenen Annahme über die relative physikalische Sicherheit des Übertragungsmediums stellt das offene Übertragungsmedium mobiler Umgebungen jedoch einen entscheidenden Einschnitt dar.

Im *Ad-hoc-Modus* hingegen werden die mobilen Geräte nur untereinander verbunden, ohne Anbindung an ein ortsfestes Netz. Somit kann nur mit Geräten innerhalb der Funkreichweite kommuniziert werden. Um auch mit Geräten außerhalb der eigenen Funkreichweite zu kommunizieren, bedarf es weiterer Protokolle zum Routing in Ad-hoc-Netzen, die im Kern alle ein kooperatives Verhalten der beteiligten Endgeräte voraussetzen.

In mobilen Umgebungen kann also die Verfügbarkeit einer hierarchisch geordneten Infrastruktur und damit auch die dedizierter Sicherheitssysteme nicht mehr vorausgesetzt werden. Somit ist auch eine zentrale Kontrolle nicht mehr ohne weiteres möglich. Die Mobilität der beteiligten Endgeräte führt außerdem zu sich praktisch permanent verändernden physikalischen Umgebungen und damit einhergehend zu dynamisch wechselnden Netztopologien. In der Folge wird es immer schwerer, klare Systemgrenzen zu ziehen, wodurch auch die für die Perimeterverteidigung notwendigen exakt festgelegten Verteidigungslinien verloren gehen. Für den Schutz mobiler Umgebungen wird daher als erstes eine Abkehr von dem bisher im Netzwerkbereich praktizierten Sicherheitsmodell vorgeschlagen:

**Postulat 1** (Fundamentaler Schutz)**.** *Der Schutz mobiler Umgebungen erfordert den bestmöglichen Schutz jedes einzelnen Endgeräts.*

Die Bedeutung des Schutzes „jedes einzelnen Glieds in der Sicherheitskette“ [Sch00] wird also wesentlich stärker betont als dies noch in herkömmlichen Netzen der Fall war. „Bestmöglicher Schutz“ umfasst dabei auch die Forderung nach mehreren, sich ergänzenden Sicherheitsmechanismen, die sowohl präventive, überwachende als auch reaktive Maßnahmen einschließen und sämtliche Schichten des Netzwerkreferenzmodells abdecken. Problematisch ist allerdings die eingeschränkte Sichtweise eines einzelnen Endgeräts auf seine Umgebung, die de facto nur bis zur Grenze seiner Funkreichweite geht. Gerade für überwachende Maßnahmen stellt diese Einschränkung ein erhebliches Manko dar, wenn es etwa um die möglichst genaue oder frühzeitige Erkennung sicherheitsrelevanter Vorfälle geht. Auch stehen die vorhandenen Defizite mobiler Endgeräte hinsichtlich Prozessorleistung, Speichergröße, Bandbreite oder auch Batterieleistung ressourcenintensiven Sicherheitsmaßnahmen entgegen. Dies führt zu der Überlegung, den Schutz der Endgeräte zwar autonom zu organisieren, ihn aber durch eine gemeinsame Grundlage entscheidend zu verbessern (siehe hierzu auch [HL03]):

**Postulat 2** (Kooperativer Schutz)**.** *Der Schutz mobiler Umgebungen lässt sich durch die gemeinsame Verwendung der vorhandenen Informationen und Ressourcen entscheidend verbessern.*

Die Verwendung verteilter Systeme im allgemeinen wie auch die Einbindung autonom und dezentral organisierter Systeme im besonderen führen jedoch zwangsläufig zu neuen Bedrohungen. Die Sicherstellung von Integrität und Vertraulichkeit der hierbei

notwendigen Kommunikation lässt sich noch relativ einfach mit herkömmlichen kryptographischen Verfahren erreichen. Auch Verfügbarkeit kann gewährleistet werden, zumindest in demselben Maß wie für die zugrunde liegende Kommunikationsebene. Durch den Wegfall zentraler Instanzen und den damit einhergehenden Verlust mehr oder weniger explizit geäußerter Annahmen zur Vertrauenswürdigkeit einzelner Subsysteme gewinnt darüber hinaus die Frage nach einer qualitativen Bewertung der übermittelten Informationen immer mehr an Bedeutung. Dies gilt insbesondere für die Integrität und die Verfügbarkeit eines Kooperationspartners: Wie kann sichergestellt werden, dass sich ein kooperierendes System wie erwartet (regelkonform) verhält, und wie können andere Systeme mit knappen Ressourcen überhaupt zur Kooperation „ermuntert“ werden? Mit anderen Worten, wie lässt sich eine vertrauenswürdige Kooperation sicherstellen? Hierbei muss auch berücksichtigt werden, dass sich mobile Endgeräte wesentlich leichter entwenden lassen als herkömmliche, wodurch Angreifer ein System unterwandern und von innen heraus attackieren können.

**Postulat 3** (Verlässliche Kooperation)**.** *Die Kooperation zum Schutz mobiler Umgebungen selbst muss ebenfalls geschützt werden. Insbesondere muss die Qualtität der Kooperation (Bereitschaft zur Kooperation, Vertrauenswürdigkeit der kooperierenden Systeme) sichergestellt werden.*

## 1.3 Mehrseitige Sicherheit

Der Gedanke, das einzelne Endgerät und damit letztlich den einzelnen Nutzer in den Vordergrund der Sicherheitsbemühungen zu stellen, ist nicht neu. So wurden seit den 90er Jahren zahlreiche Arbeiten zur Dezentralisierung der IT-Sicherheit veröffentlicht. Herausragend ist hierbei der Gedanke der sogenannten *mehrseitigen Sicherheit*, der die Einbeziehung der Schutzinteressen *aller* Beteiligten beim Aufbau einer Kommunikationsverbindung unter Austragung der daraus resultierenden Interessenskonflikte anstrebt [FP97]. Partner einer mehrseitig sicheren Kommunikationsbeziehung interagieren also in einem ausgewogenen Kräfteverhältnis miteinander. Auch kann die Realisierung mehrseitiger Sicherheit gegenläufige und miteinander unvereinbare Interessen der Beteiligten offenbaren [Fed98].

Dabei ist mehrseitige Sicherheit nicht als Technik oder Verfahren, sondern vielmehr als eine Handlungsanleitung für die Gestaltung von Kommunikationssystemen zu verstehen, deren Anwendung sich derzeit allerdings fast ausschließlich auf präventive Sicherheitsmechanismen konzentriert [Pfi06]. Die Bedeutung des damit verbundenen Paradigmenwechsels für die Netzsicherheit geht jedoch weiter, wie Gattung et al. in [GGPS97] recht treffend bemerken :

> *Während in den klassischen Sicherheitsmodellen die Sicht des Netzbetreibers im Vordergrund steht, berücksichtigt die mehrseitige Sicht auch und gerade die Sicht*

*der Benutzer. Neben dem Schutz, den die Netze ihren Benutzern bieten, werden die Benutzer auch Mechanismen zum Selbstschutz benötigen, um die Abhängigkeit von anderen zu reduzieren.*

Die im vorangegangenen Abschnitt aufgestellten Forderungen folgen dieser Sichtweise. Dies ist nur konsequent, verschwimmt doch in mobilen Umgebungen zusehends die klare Unterscheidung zwischen Netzbetreiber und -benutzern. So stellen die Nutzer mobiler Ad-hoc-Netze gemeinsam den Netzbetreiber, indem sie beim Aufbau und Betrieb der notwendigen „Infrastruktur“ zusammenarbeiten.

Die vorliegende Arbeit kann daher als Erweiterung des Gedankens der mehrseitigen Sicherheit auf reaktive Sicherheitsmechanismen gesehen werden. Anstatt des heute üblichen Ansatzes einer verteilten Überwachung von Netzen mit einer zentralen Auswertung soll jedes Endgerät individuell den aktuellen Sicherheitszustand bestimmen. Um die Vielzahl der Einzelsichten zu objektivieren oder gar einen allgemein akzeptierten Sicherheitszustand abzuleiten, müssen die einzelnen Endgeräte miteinander kooperieren. Widersprüchliche Sichtweisen sind dabei aufzulösen. Hierfür ist eine Vertrauensbasis zwischen den Kooperationspartnern erforderlich [MP97a]. Diese lässt sich jedoch nur dann aufrecht erhalten, wenn die einzelnen Partner gemäß der an sie gestellten Erwartungen agieren.

## 1.4 Beiträge und Gliederung der Arbeit

Nach diesem einleitenden Plädoyer für einen geräte- und nutzerzentrierten Schutz mobiler Umgebungen, der neben präventiven auch überwachende sowie reaktive Sicherheitsmechanismen umfassen soll, wird das Problemfeld im Verlauf der Arbeit weiter konkretisiert.

Kapitel 2 analysiert dazu die Eigenschaften mobiler Umgebungen und darin denkbarer Angriffe, um *Anforderungen an eine mobile Sicherheitsarchitektur* zu identifizieren. Diese dienen dann als Kriterien zur Bewertung existierender Ansätze. Im weiteren Verlauf konzentrieren sich die Betrachtungen auf den Übertragungsstandard IEEE 802.11. Die hierbei durchgeführte Recherche stellt den *ersten umfassenden Vergleich wissenschaftlicher und kommerzieller Ansätze zum Schutz von Wireless LANs* dar. Die untersuchten Arbeiten werden dabei in die Kategorien präventive, reaktive sowie angriffstolerante Sicherheitsarchitektur eingeteilt. Im Ergebnis bestätigt sich die anfängliche Einschätzung, dass nur die Kombination präventiver und reaktiver Mechanismen einen umfassenden Schutz ermöglicht. Dieser Umstand wurde bereits durch eine Vielzahl akademischer Publikationen zur Angriffserkennung in mobilen Netzen gewürdigt. Gerade im Bereich mobiler Ad-hoc-Netze fehlen jedoch aussagekräftige Untersuchungen der vorgeschlagenen Verfahren. Die üblicherweise durchgeführten Simulationen weisen eine Vielzahl von Problemen auf, z. B. unrealistische Annahmen über das Bewegungsverhalten von

Knoten [PM07] oder zur Systemumgebung und Signalausbreitung [KNG+04]. Darüber hinaus weisen einige Arbeiten handwerkliche Mängel auf, wie etwa die ungenügende statistische Auswertung mehrerer unabhängiger Simulationsläufe [AY06]. All dies lässt einen integrierten Ansatz sinnvoll erscheinen, der neben einer simulativen Betrachtung auch die experimentelle Untersuchung im realen oder emulierten Umfeld einschließt [CSS02]. Die vorliegende Arbeit setzt sich die Entwicklung der dazu notwendigen Systemumgebung zum Ziel [Gro06a, Gro06b].

Der Entwurf der Rahmenarchitektur für eine solche *praxisbezogene Systemumgebung zum Schutz mobiler Endgeräte* wird in Kapitel 3 vorgestellt. Wesentlicher Grundsatz bei ihrer Entwicklung war die Trennung des eigentlichen Sicherheitsmechanismus von der Steuerung durch den Nutzer. Beide werden über einen separaten Controller miteinander gekoppelt. Dieser koordiniert auch die Kooperation mit anderen Geräten, wozu er auf gesonderte Mechanismen zum Auffinden und Auswählen von Kooperationspartnern zurückgreift.

Die Realisierung der vorgeschlagenen Systemarchitektur im Rahmen eines Prototyps zur *kooperativen Erkennung von Angriffen auf Wireless LANs* wird in den beiden folgenden Kapiteln erläutert.

So beschreibt Kapitel 4 die Entwicklung eines *Intrusion Detection Systems für Wireless LANs*, das lokal auf einem Endgerät anhand empfangener Netzwerkpakete mögliche Angriffe identifiziert [GN06]. Der Systementwurf trennt dabei strikt zwischen der Netzwerkanalyse und der daraus resultierenden Generierung von Ereignissen sowie ihrer Interpretation, so dass eine einfache Anpassung der zugrunde gelegten Sicherheitsrichtlinien ermöglicht wird. Um der in Kapitel 1.2 aufgestellten Forderung nach Zusammenarbeit zwischen mobilen Endgeräten bei der Angriffserkennung nachzukommen, unterstützt das entwickelte Wireless IDS auch den Austausch von Ereignismeldungen und die gegenseitige Warnung vor erkannten Angreifern. Dabei wird zwischen zwei Ansätzen unterschieden. Für überschaubare oder zentral administrierte Umgebungen lassen sich mobile Knoten zu vorkonfigurierten engen Kooperationsgruppen zusammenschließen. Dieser Ansatz skaliert jedoch mit steigender Anzahl von Kooperationspartnern und in den offenen Umgebungen mobiler Ad-hoc-Netze immer schlechter. Deshalb wird eine Möglichkeit zur dynamischen Konfiguration einer selbstorganisierenden verteilten Angriffserkennung benötigt, mit der geeignete Kooperationspartner ausgewählt und die anfallenden Aufgaben untereinander aufgeteilt werden können.

Kapitel 5 beschreibt daher die *Realisierung einer Kooperations-Middleware unter Verwendung von Peer-to-Peer-Technologien*. Da eine Kooperation für ein Endgerät grundsätzlich mit einem erhöhtem Ressourcenverbrauch verbunden ist, wird außerdem ein Ansatz zur Auswahl möglichst geeigneter Kooperationspartner vorgestellt. Als Auswahlkriterien werden exemplarisch die geographische Position eines potentiellen Kooperationspartners sowie das in ihn gesetzte Vertrauen angeführt. Die hierzu notwendigen Informationen werden durch externe Systeme ermittelt. Für ihre Anbindung an das Framework spezifiziert der Systementwurf geeignete Schnittstellen. Die Validierung der entwor-

fenen Kooperations-Middleware erfolgt mittels einer im Rahmen einer Diplomarbeit [Rei07] durchgeführten prototypischen Implementierung. Durch die in den Kapiteln 4 und 5 beschriebenen Arbeiten wird damit ein praxistaugliches *Experimentierumfeld zur Erprobung reaktiver Sicherheitsmechanismen* in mobilen Netzen bereitgestellt.

Um diese Systemumgebung auch außerhalb des Labors sinnvoll nutzen zu können, bedarf es jedoch der weiteren Ausgestaltung der bereits erwähnten Auswahlkriterien für mobile Kooperationspartner, um insbesondere Angriffe auf die vorgeschlagene Systemlösung einzudämmen. Von besonderem Interesse ist hierbei die *Einschätzung der Verlässlichkeit kooperierender Knoten*, d. h. die Bewertung ihrer Kooperationsbereitschaft und der Qualität der von ihnen bereitgestellten Leistungen als *semantische Erweiterung herkömmlicher Schutzziele*. Die Erfassung der Verlässlichkeit eines Endgeräts ermöglicht dabei die dynamische Anpassung des ihm entgegengebrachten Vertrauens.

In Internetcommunities, wie Online-Foren, elektronischen Marktplätzen oder elektronischen sozialen Netzen, haben sich in der Vergangenheit Reputationssysteme als probates Mittel zur Einschätzung der Verlässlichkeit ihrer Teilnehmer erwiesen. Reputationssysteme sammeln die Erfahrungen, die Nutzer im Rahmen ihrer Interaktionen mit anderen Nutzern einer Internetcommunity machen. Die Reputation, die sich ein Nutzer auf diese Weise erarbeitet, liefert anderen einen Anhaltspunkt für sein bisheriges Verhalten und somit eine Grundlage zur Einschätzung des von ihm zu erwartenden Verhaltens in neuen Interaktionen.

Mit der wachsenden Verbreitung offener verteilter Systeme, wie Peer-to-Peer-Netzen oder mobilen Ad-hoc-Netzen, ist auch die Frage nach deren Sicherheit immer stärker ins Bewusstsein gerückt. Genau wie Internetcommunities basieren diese Umgebungen auf der Interaktion zwischen einander zuvor unbekannten und nur partiell vertrauenden Entitäten. Es liegt daher nahe, auch hier die Verwendung von Reputationssystemen zu ihrem Schutz zu diskutieren. Genau dies tun erste Arbeiten in den Bereichen Peer-to-Peer-Netze und Ad-hoc-Routing, die aber die konkrete Konstruktion des verwendeten Reputationssystems meistens weitgehend ausblenden. Aus den bereits in Kapitel 1.2 erläuterten Gründen ist aber der Einsatz heute üblicher zentralisierter Ansätze für Reputationssysteme problematisch.

Kapitel 6 gibt daher eine systemorientierte Einführung in Reputationssysteme und analysiert die Einschränkungen mobiler Umgebungen, um daraus *Entwurfsoptionen für ein mobiles Reputationssystem* abzuleiten. Ziel ist dabei nicht die Konzeption und Validierung einer umfassenden Systemlösung, was den Rahmen der Arbeit bei weitem sprengen würde, sondern die Systematisierung und Präzisierung der konkreten Fragestellung sowie eine daran anschließende Diskussion einzelner Teilaspekte als Grundlage für weitergehende Arbeiten.

# 2
# Sicherheit in mobilen Umgebungen

Im folgenden wird die Einsatzumgebung der geplanten Sicherheitsarchitektur analysiert. Hierzu werden zunächst die Termini mobiles Endgerät respektive drahtloses Netz eingeführt sowie ihre spezifischen Eigenschaften beschrieben. Daraus werden Anforderungen für den Schutz mobiler Umgebungen abgeleitet. In der Folge wird eine kurze Einführung in Intrusion Detection Systeme als zentralem Bestandteil einer umfassenden Sicherheitsarchitektur gegeben, um dann für bereits existierende Lösungen zu untersuchen, ob und wie sie die beschriebenen Anforderungen erfüllen. Das Kapitel schließt mit einer Diskussion der gewonnenen Erkenntnisse zur Motivation der weiteren Zielsetzung.

## 2.1 Eigenschaften mobiler Umgebungen

Nach Roth ermöglicht ein *mobiles Endgerät* einem Benutzer die Verwendung lokal oder mittels eines mobilen Netzes verfügbarer Anwendungen und Dienste [Rot05a]. Ein *mobiles Netz* bezeichnet dabei üblicherweise die drahtlose Vernetzung mobiler Endgeräte untereinander sowie mit stationären Netzen. Die drahtlose Vernetzung mobiler Endgeräte untereinander erfolgt mittels sogenannter *Ad-hoc-Vernetzung*, worunter kurzfristig und spontan initiierte drahtlose Verbindungen zwischen Endgeräten ohne feste Kommunikationsinfrastruktur verstanden werden. Die Verbindung zu stationären Netzen erfolgt über eine dedizierte Netzinfrastruktur, sogenannte Access Points. Roth unterstreicht zudem die im Vergleich zu konventionellen Geräten deutlich persönlichere Natur der Verwendung mobiler Endgeräte und spricht hierbei von *Personal Computing*.

### 2.1.1 Taxonomie mobiler Endgeräte

Detailliertere Klassifikationen mobiler Endgeräte sind in der Literatur kaum zu finden. Roth unterteilt mobile Endgeräte auf Grundlage ihrer Hardwareplattform in fünf Kategorien:

*Mobile Standardcomputer:* Hierunter werden mobile Geräte mit praktisch derselben Leistungsfähigkeit wie ihre herkömmlichen stationären Pendants zusammengefasst.

*Bordcomputer:* Dabei handelt es sich um fest in Fahrzeugen installierte Geräte zur Erfüllung spezieller Aufgaben, wie etwa Navigationssysteme.

*Handhelds:* Hierunter werden alle mobilen Geräte zusammengefasst, die klein genug sind, um in einer Hand gehalten und im gehaltenen Zustand bedient zu werden. Im Gegensatz zu mobilen Standardcomputern ist ihr Leistungsumfang jedoch meist deutlich reduziert.

*Wearables:* Wearables sind Endgeräte, die nicht mehr in der Hand gehalten, sondern am Körper getragen werden.

*Chipkarten und Etiketten:* Chipkarten schließlich sind eine Variante kleinster mobiler Geräte mit deutlich reduziertem Leistungsumfang, die zur Bedienung ein spezielles Lesegerät benötigen. Sie stellen damit strenggenommen kein selbständiges Endgerät dar, verfügen aber über Speicher und Prozessor und können teilweise auch programmiert werden. Sie werden hauptsächlich zur Identifizierung eingesetzt. Zur selben Kategorie gehören auch intelligente Etiketten, wie etwa RFID-Transponder. Ein passiver RFID-Transponder (Radiofrequenz-Identifikation) umfasst einen Prozessor, Speicher und eine Antenne, verfügt jedoch über keine eigene Stromversorgung. Die zum Arbeiten notwendige Energie wird stattdessen aus den an sie gerichteten Funksignalen generiert. RFID-Transponder werden beispielsweise in der Logistik zur Transportverfolgung eingesetzt oder aber als kontaktlos auslesbarer Datenspeicher in modernen Reisepässen.

Orthogonal hierzu unterscheidet Roth außerdem zwischen *Universal-* und *Spezialgeräten*, je nachdem ob ein Endgerät für einen dedizierten Zweck konzipiert wurde oder aber offen für quasi beliebige Anwendungen ist.

Wie Roth selbst einräumt, gestaltet sich eine fundierte Klassifizierung mobiler Endgeräte schwierig. Dies spiegelt sich auch in den von ihm gewählten Gerätekategorien wider. So orientiert sich beispielsweise die Einteilung in mobile Standardcomputer an der Leistungsfähigkeit des Endgeräts. Eine solche Einordnung ist jedoch, wie Roth selbst anmerkt, dem permanenten technologischen Wandel unterworfen und deshalb nur bedingt aussagekräftig. Ferner ist die Kategorisierung nicht eindeutig. So gibt es per

Definition keine universellen Bordcomputer und leistungsstarke Handhelds könnten durchaus auch den mobilen Standardcomputern zugerechnet werden. Man kann davon ausgehen, dass solche funktionalen Überlappungen und Variationen innerhalb der Geräteklassen in Zukunft noch zunehmen werden [Wie04]. In der Praxis haben sich derartige Klassifikationen nach Geräteklassen dennoch durchgesetzt.

Verbreitet ist auch die Kategorisierung mobiler Endgeräte nach der verwendeten Softwareplattform, maßgeblich dem eingesetzten mobilen Betriebssystem. Stellvertretend seien hier PalmOS, Windows CE, Symbian OS und vermehrt auch Linux genannt. Da sich die in dieser Arbeit vorgeschlagene Architektur auf die Sicherheit der Netzverbindung konzentriert und von der verwendeten Betriebssystemumgebung abstrahiert, sollen die Besonderheiten der einzelnen Systeme hier nicht näher erörtert werden. Für einen genaueren Überblick wird stattdessen auf [Rot05a] verwiesen.

### 2.1.2 Taxonomie drahtloser Netze

Ein übliches Kriterium zur Klassifizierung von Netzen stellt deren Abdeckung dar. Analog zu Festnetzen kann damit zwischen drahtlosen Fern- (engl. *Wide Area Network – WAN*), Stadt- (engl. *Metropolitan Area Network – MAN*) und lokalen Netzen (engl. *Local Area Network – LAN*) unterschieden werden, wobei die drahtlosen lokalen Netze nochmals in *Wireless Local Area Networks (WLANs)* und *Wireless Personal Area Networks (WPANs)* unterteilt werden. Im Gegensatz zu WLANs dienen WPANs der Überbrückung kürzester Distanzen und waren ursprünglich als Ersatz für fliegend verlegte Kabelverbindungen bei der Kopplung mobiler Kleingeräte oder der Anbindung von Peripheriegeräten und Sensoren gedacht. Hierzu gehören neben den als Bluetooth bekannten und im Standard IEEE 802.15.1 definierten Pico-Netzen auch Ultra Wide Band Technologien nach IEEE 802.15.3 zur Realisierung möglichst hoher Übertragungsraten und IEEE 802.15.4 – auch bekannt als ZigBee – zur Steigerung der Energieeffizienz. Mittlerweile definiert Bluetooth aber auch sogenannte Klasse-1-Geräte, die eine Reichweite bis etwa 100 Meter haben und damit die ursprünglich anvisierten Distanzen deutlich übertreffen. Ein weiteres Kriterium stellt die Unterscheidung hinsichtlich der maximal möglichen Bandbreite und der daraus resultierenden Datenübertragungsrate dar. Diese war anfang noch sehr begrenzt. So ermöglichte etwa der GSM-Standard eine maximale Übertragungsrate von lediglich 9,6 Kbps. Jüngere Entwicklungen, wie etwa UMTS oder auch IEEE 802.11n, ermöglichen hingegen wesentlich größere Bandbreiten, so dass ihre Übertragungsraten deutlich im Megabit-Bereich liegen. Abbildung 2.1 auf der nächsten Seite zeigt eine Übersicht heute gängiger drahtloser Netzwerkstandards, geordnet nach Abdeckung und maximal möglicher Datenrate (vgl. auch [KR07a]).

Eine weitere Möglichkeit der Klassifizierung mobiler Netze stellt die Unterscheidung hinsichtlich der verwendeten Netzarchitektur dar [KR07a]. Abstrakt wird hier zwischen infrastrukturlosen und infrastrukturbasierten Netzen sowie Single-Hop- und Multi-Hop-Netzen unterschieden, je nachdem ob Pakete innerhalb eines drahtlosen Netzes exakt an einen oder mehrere Knoten weitergeleitet werden können respek-

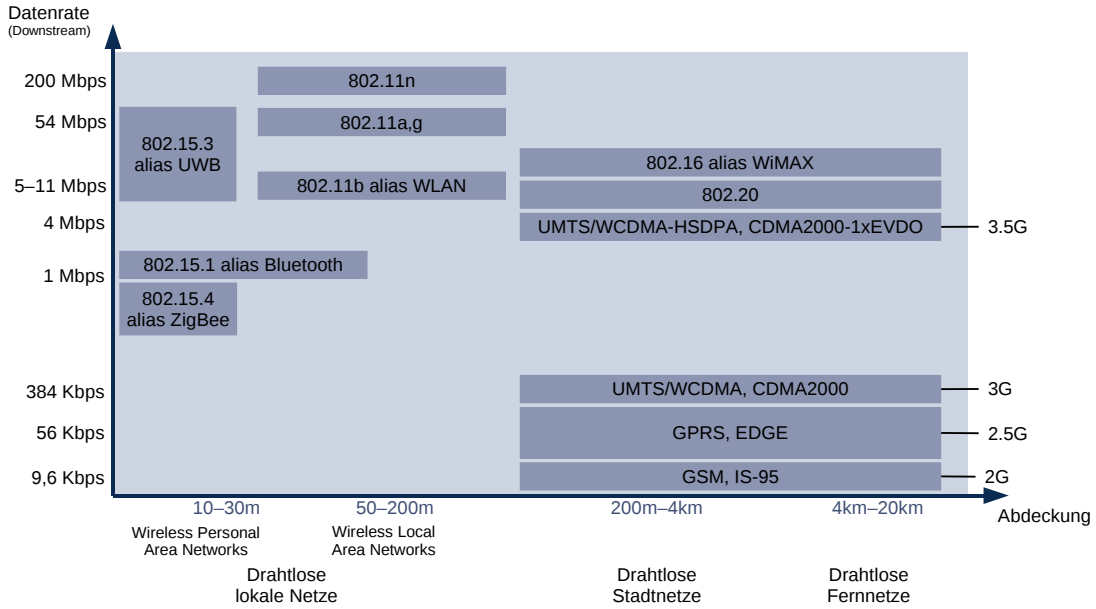



Abbildung 2.1: Klassifikation drahtloser Datennetze nach Datenrate und Abdeckung

tive koordinierende Basisstationen integraler Bestandteil der Netzinfrastruktur sind oder nicht. Abbildung 2.2 auf der nächsten Seite zeigt eine Einordnung ausgewählter Netzwerkstandards in die genannten Klassen.

- *Infrastrukturbasierte Single-Hop-Netze* sind dabei über eine Basisstation mit einem größeren drahtgebundenen Netzwerk verbunden. Die Basisstation koordiniert zudem den gesamten Datenverkehr innerhalb des drahtlosen Netzes, d.h. alle Clients des drahtlosen Netzes sind über eine drahtlose Verbindung direkt mit der Basisstation gekoppelt. Jegliche Kommunikation zwischen den Clients innerhalb des mobilen Netzes erfolgt über die Basisstation. Die zellularen Mobilfunknetze der zweiten bis dritten Generation sind ebenso Beispiele für drahtlose Netze dieser Kategorie, wie im Infrastrukturmodus betriebene Wireless LANs nach IEEE 802.11 Standard oder aber auch die neuen WiMAX-Netze gemäß IEEE 802.16.

- In *infrastrukturlosen Single-Hop-Netzen* fehlt diese zentrale Basisstation und damit eine Anbindung an ein nachgelagertes drahtgebundenes Netzwerk. Stattdessen sind die Endgeräte des Netzes untereinander verbunden, wobei Daten immer nur direkt zwischen zwei Knoten ausgetauscht werden können. Netze dieser Kategorie stellen damit einen Extremfall dar, da sie sowohl auf jegliche zentrale Koordination verzichten als auch nur rudimentäre Kommunikationsmöglichkeiten

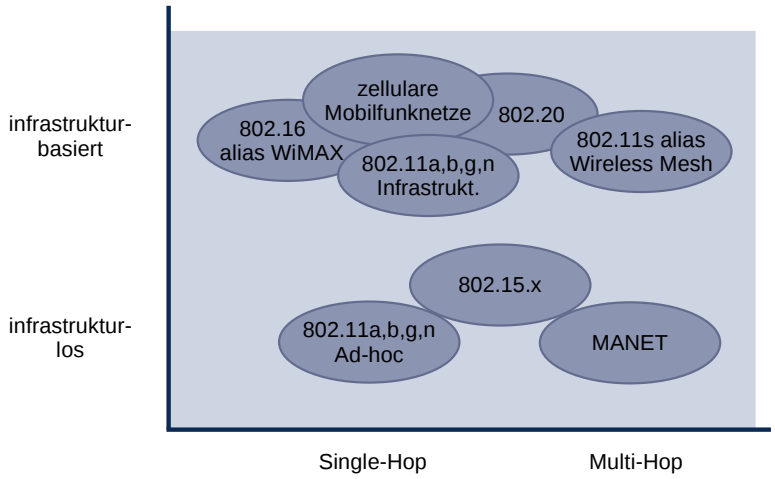



Abbildung 2.2: Klassifikation drahtloser Datennetze hinsichtlich ihrer Netzarchitektur

bieten. Neben im Ad-hoc-Modus betriebenen Wireless LANs nach IEEE 802.11 Standard wird auch der IEEE 802.15.1 Standard zu den infrastrukturlosen Single-Hop-Netzen gezählt, obwohl bei diesem ein Master-Knoten eine koordinierende Rolle einnimmt, indem er die Sendeslots an die Slaves eines Pico-Netzes vergibt. Im Gegensatz zu einer zentral vorgegebenen Basisstation wird der Bluetooth-Master jedoch dynamisch bestimmt. Durch den Zusammenschluss von bis zu zehn Pico-Netzen zu einem sogenannten Scatter-Netz wird prinzipiell sogar eine Multi-Hop-Kommunikation unterstützt. Die Bildung solcher Scatter-Netze ist jedoch bis heute nicht standardisiert.

- Die *infrastrukturbasierten Multi-Hop-Netze* repräsentieren das zu den infrastrukturlosen Single-Hop-Netzen gegenläufige Extrem. In diesen Netzen verbindet sich jedes einzelne mobile Endgeräte mit einer zentralen Basisstation. Zur Vergrößerung der Funkabdeckung sind diese wiederum über einen funk- oder kabelbasierten Backbone, dem sogenannten *Distribution System*, untereinander gekoppelt, um Datenpakete der Endgeräte weiterzuleiten. In diese Kategorie fallen beispielsweise *Wireless Mesh Netze*, wie sie etwa im kommenden Standard IEEE 802.11s definiert werden.

- Damit verbleiben als letzte Klasse die *infrastrukturlosen Multi-Hop-Netze*, die auf zentrale Basisstationen verzichten. Stattdessen vermitteln die einzelnen Knoten eingehende Nachrichten untereinander. Durch die fehlende Abhängigkeit von einer zentralen Infrastruktur sind Netze dieses Typs auch deutlich besser für mobile Knoten geeignet. Beispielhaft seien hier *mobile Ad-hoc-Netze (MANETs)* etwa auf Grundlage des IEEE 802.11 Standards genannt.

## 2.2 Angriffe auf mobile Umgebungen

Wie alle Netze müssen auch mobile Umgebungen vor möglichem Missbrauch geschützt werden. Ein solcher Missbrauch manifestiert sich in einer Folge von Handlungen, *Angriff* genannt, die die Funktionsfähigkeit oder Sicherheit des mobilen Netzes beeinträchtigen können.

Wie bereits in Kapitel 1.4 auf Seite 7 dargelegt, strebt die vorliegende Arbeit nach einem weitgehenden Schutz vor solchen Angriffen. Dafür ist es hilfreich, zunächst eine möglichst große Anzahl grundsätzlich denkbarer Angriffe strukturiert zu erfassen, um so potentielle Bedrohungen systematisch zu ermitteln. Hierbei wird im Folgenden zwischen Angriffen in infrastrukturlosen und solchen in infrastrukturbasierten Netzen unterschieden.

Zur Darstellung und Analyse der dabei auftretenden Angriffsformen werden sogenannte Bedrohungs- oder *Angriffsbäume* (engl. *Attack Trees*) verwendet. Angriffsbäume sind ein hierarchisches Verfahren, das Angriffsszenarien strukturiert in Form von Bäumen mit dem Angriffsziel als Wurzelknoten und den unterschiedlichen Angriffen entlang der einzelnen Äste modelliert [Sch99]. Jeder Knoten repräsentiert dabei ein zu erfüllendes Teilziel. Gibt es mehrere Handlungsalternativen, dieses Ziel zu erreichen, werden diese eine Ebene tiefer aufgeführt und mit OR gekennzeichnet. Sind zum Erreichen des Ziels mehrere Schritte auszuführen, wird stattdessen die Bezeichnung AND verwendet. Außerdem können die einzelnen Knoten mit Attributen versehen werden, um etwa die Kosten oder Eintrittswahrscheinlichkeiten von Teilangriffen zu berücksichtigen.

Die reine Erfassung möglicher Bedrohungen bzw. Angriffe ist jedoch nur ein Schritt bei der Konstruktion sicherer Systeme. Um sinnvolle Sicherheitsanforderungen definieren zu können, müssen die Angriffe außerdem bewertet werden, etwa hinsichtlich der verursachten Schäden oder ihrer Eintrittswahrscheinlichkeit [Eck03]. Hierfür ist eine genauere Betrachtung des potentiellen Angreifers in Form eines sogenannten *Angreifermodells* von Nöten, das die Stärke eines Angreifers bestimmt. Umgekehrt kann dieses auch zur Charakterisierung vorhandener Sicherheitsmechanismen verwendet werden.

In der Literatur finden sich viele Herangehensweisen zur Erstellung solcher Angreifermodelle. Dabei werden je nach Standpunkt der Autoren einzelne Aspekte eines Angreifers betont und andere vernachlässigt. Neben der *Rolle* des Angreifers (z. B. Innen- oder Außentäter) und seiner Verbreitung beschreibt ein Angreifermodell in der Regel auch sein *Verhalten* (passiv/beobachtend versus aktiv/eingreifend) und die ihm zu Verfügung stehenden *Ressourcen*, wie etwa Geld, Zeit oder Know-how [Pfi00]. Ein Angreifer wird ferner durch seine *Motivation* (z. B. Neugier, Vergnügen oder Vorteilsnahme) und die von ihm verfolgten *Ziele* (z. B. Störung des Netzes, Zugriff auf Informationen oder Veränderung von Daten) charakterisiert. Eine weitere Einteilung ergibt sich durch die von ihm verwendeten *Techniken*, beispielsweise Maskerade oder Abhören. In der Praxis fallen konkrete Angriffe meist in mehrere der genannten Kategorien, so dass

eine solche Einteilung immer nur als grober Orientierungspunkt beim Entwurf einer Sicherheitsarchitektur dienen kann.

Im folgenden werden nun potentielle Bedrohungen für mobile Umgebungen untersucht. Hierbei wird zunächst ein Angreifermodell aufgestellt, bevor am Beispiel von mobilen Ad-hoc-Netzen und von Wireless LANs genauer auf die Ziele der einzelnen Angreifertypen eingegangen wird. Die Konzentration auf MANETs und WLANs ist einerseits der Verbreitung des ihnen zugrunde liegenden Standards IEEE 802.11 geschuldet, andererseits aber auch notwendig, um die Angriffe fokussiert diskutieren zu können. Zumindest die oberen Ebenen der resultierenden Angriffsbäume lassen sich jedoch allgemein an beliebige mobile Netze mit und ohne dedizierter Infrastruktur anpassen.

### 2.2.1 Angreifermodell

Angreifer in mobilen Umgebungen werden meist nach ihrem Verhalten oder ihrer Motivation klassifiziert (siehe beispielsweise [HP04, Kar03]). Beobachten sie lediglich den Datenfluss ohne aktiv in eine mobile Umgebung einzugreifen, sind sie praktisch nicht zu erkennen. Jedoch lassen sich die Erfolgsaussichten eines solch einfachen *Sniffing*-Angriffs durch verschiedene präventive Maßnahmen deutlich reduzieren oder sogar gänzlich ausschließen. Um diese Maßnahmen zu umgehen, ist ein Angreifer gezwungen, selbst aktiv zu werden. Damit setzt er sich der Gefahr aus, entdeckt zu werden. Die sich aus einer Entdeckung für ihn ergebenden Konsequenzen sind in der Praxis jedoch relativ gering. So ist die Erkennung eines Angriffs in einer mobilen Umgebung mitnichten gleichzusetzen mit der Lokalisierung oder gar Identifizierung eines mobil agierenden Angreifers. Es besteht also kaum ein Grund, aktive Angriffe zu meiden, weshalb das Angreiferverhalten nur bedingt zur Einschätzung der potentiellen Gefährdung durch einen Angriff taugt.

Stattdessen erscheint es realistischer, hierfür die Motivation eines Angreifers heranzuziehen. Im Gegensatz zu klassischen Datennetzen müssen im mobilen Umfeld dabei nicht nur böswillig agierende Netzknoten berücksichtigt werden, sondern auch solche, die fehlerhaft arbeiten oder egoistische Beweggründe für ihr Handeln haben. Kargl fasst dies unter dem Begriff *FEB-Knoten* zusammen [Kar03].

*Fehlerhafte Knoten* weisen eine Fehlfunktion ausgelöst durch eine unzulängliche Implementierung oder Konfiguration auf. Dies wird oft auch als byzantinisches Verhalten bezeichnet [LSP82]. Beispiele fehlerhafter Knoten sind Access Points, die legitime Nutzer ausschließen, oder aber mobile Knoten, die nicht mehr ordnungsgemäß am Ad-hoc-Routing-Verfahren teilnehmen. Auch wenn beide keine Angreifer im strengen Sinne darstellen, so sind die von ihnen verursachten Auswirkungen auf eine mobile Umgebung durchaus mit denen vorsätzlich agierender Angreifer vergleichbar, weshalb sie hier ebenfalls aufgeführt werden.

*Egoistische Knoten* verhalten sich insofern rational, als sie nach der Maximierung ihres persönlichen Nutzens streben. Dies impliziert ein prinzipielles Interesse an der Funktionsfähigkeit des mobilen Netzes. Sie trachten jedoch danach, ihren persönlichen Beitrag hierzu möglichst gering zu halten. Bezogen auf infrastrukturlose mobile Umgebungen bedeutet dies, dass ein egoistischer Knoten die gemeinsam gebildete Netzbasis zwar nutzen, gleichzeitig aber möglichst wenig eigene Ressourcen für deren Erhaltung beisteuern will. Bei infrastrukturbasierten Netzen geht es hingegen darum, über das mobile Netz Zugang zu dem nachgelagerten Weitverkehrsnetz zu erhalten. Außer den hierfür benötigten Ressourcen wird die mobile Umgebung nicht weiter gestört.

*Böswillige Knoten* nehmen im Gegensatz zu egoistischen Knoten keinerlei Rücksicht auf die Funktionsfähigkeit einzelner Knoten oder des Netzes als Ganzes. Sie streben ausschließlich nach dem Erreichen ihres Angriffsziels, das beispielsweise auch im vollständigen Lahmlegen der mobilen Umgebung bestehen kann.

### 2.2.2 Angriffsziele

Im folgenden werden nun die möglichen Ziele der einzelnen Angreifertypen getrennt nach mobilen Ad-hoc-Netzen und Wireless LANs genauer spezifiziert. Abbildung 2.3 zeigt einen Überblick der grundsätzlichen Angriffspunkte, wobei die für mobile Umgebungen maßgeblichen dunkel gezeichnet sind. In Anlehnung an das ISO/OSI-Modell wird hierbei grob zwischen Angriffen auf die unteren Schichten inklusive des physischen Übertragungsmediums und solchen auf die „Infrastruktur“ eines mobilen Netzes selbst unterschieden. Diese manifestieren sich in Netzen mit und ohne dedizierter Infrastruktur in der Manipulation der Wegewahl oder der Paketweiterleitung. Den höheren Schichten zugeordnet wird die direkte Beeinflussung mobiler Endgeräte, die im wesentlichen auf der Umgehung von Schutzmaßnahmen des Betriebssystems oder der Geräte-Hardware beruht.

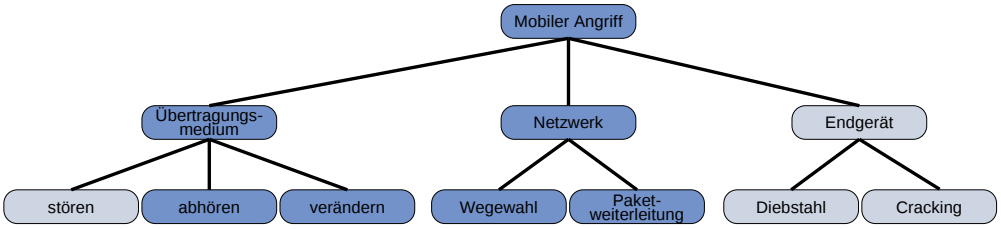



Abbildung 2.3: Angriffspunkte in mobilen Umgebungen

Die Betrachtung von Angriffen in mobilen Ad-hoc-Netzen konzentriert sich auf das Ad-hoc-Routing als ihr charakterisierendes Element. Das Abhören (engl. *Spoofing*) oder

die Manipulation des Übertragungsmediums durch Veränderung der übermittelten Informationen respektive Senden eines Störsignals (engl. *Jamming*) ist ebenso wenig spezifisch für mobile Ad-hoc-Netze wie die Übernahme eines einzelnen Knotens durch sogenanntes *Cracking* [SR96] oder durch simplen Diebstahl des mobilen Endgeräts. Auf beides wird im Folgenden daher nicht näher eingegangen.

Im Gegensatz zu mobilen Ad-hoc-Netzen verfügen Wireless LANs mit ihren Access Points hingegen über dedizierte Hardware zum Aufbau der Netzstruktur. Über sie wird der Netzzugang für die einzelnen mobilen Endgeräte gebündelt und beschränkt. Sämtliche Angriffe in Wireless LANs zielen daher im Kern auf den Übertragungsweg zwischen Endgerät und Access Point.

#### Ziele egoistischer Angreifer

Während fehlerhafte Knoten per Definition kein Angriffsziel verfolgen, streben egoistische Knoten in mobilen Ad-hoc-Netzen danach, möglichst wenig Ressourcen für die Aufrechterhaltung der Netzfunktionalität aufzuwenden:

*Einsparen eigener Ressourcen:* Bezogen auf das Ad-hoc-Routing-Protokoll bedeutet ein solcher auch *Selfishness* genannter Angriff, dass entweder die Wegewahl oder die Paketweiterleitung sabotiert wird. Besonders kritisch sind hierbei Angreifer, die zwar korrekt an der Wegewahl teilnehmen, anschließend aber die Weiterleitung von Paketen verweigern. Simulationen haben gezeigt, dass gängige Ad-hoc-Routing-Protokolle nicht in der Lage sind, dieses Verhalten zu erkennen, um mit der Wahl einer alternativen Route zu reagieren [Kar03]. Egoistische Angreifer beeinträchtigen somit die Verfügbarkeit und Integrität eines MANETs.

Egoistische Angreifer in Wireless LANs versuchen stattdessen über das mobile Netz Zugang zu den ihm nachgelagerten Weitverkehrsnetz zu erlangen:

*Zugang zum Netz:* Bevor überhaupt Zugang zu einem WLAN erlangt werden kann, muss ein Angreifer erst einmal von dessen Existenz erfahren. Dies kann er entweder durch passives Abhören des Funkverkehrs erreichen oder aber mittels einer aktiven Suche, bei der er permanent sogenannte *Probe Requests* aussendet und auf eine entsprechende *Probe Response* des Access Points wartet. Arbeitet das WLAN unverschlüsselt, kann er nun meist direkt darauf zugreifen, indem er sich am Access Point anmeldet. Bei verschlüsselten Netzen benötigt er hierfür noch den verwendeten Schlüssel, der sich je nach verwendetem Sicherheitsprotokoll entweder durch vollständige Suche oder aber aus dem mitgehörten Netzverkehr ermitteln lässt. Zwar sollte eine vollständige Suche durch die Wahl eines geeignet großen Schlüsselraumes unmöglich sein, in der Vergangenheit gelang es jedoch immer wieder, diesen durch Ausnutzung von Protokollschwächen hinreichend zu verkleinern.

### Ziele böswilliger Angreifer

Böswillige Knoten können darüber hinaus weitere Zielsetzungen verfolgen, die sich allgemein auf die Verletzung eines oder mehrerer der klassischen Schutzziele der Informationssicherheit zurückführen lassen:

*Beeinträchtigung der Funktionsfähigkeit des Netzes:* Sowohl durch die Überlastung einzelner Netzknoten als auch durch die Störung der Routing-Funktionalität können signifikante Teile eines MANETs lahmgelegt werden. Ersteres lässt sich durch das Versenden unnötiger oder gefälschter Datenpakete bewerkstelligen. Ein Beispiel hierfür ist ein sogenannter *Discovery Storm*, bei dem unnötige Route Requests an beliebige Knoten eines MANETs verschickt werden. Ein *Black-Hole-Routing*-Angriff führt ebenso zu einer Störung der Routing-Funktionalität wie eine *Cache-Pollution*-Attacke. Während erstgenannter dafür sorgt, dass Pakete im Netz verloren gehen, werden bei letzterer Topologiedaten zerstört. Im Ergebnis trachten somit alle genannten Verfahren danach, die Verfügbarkeit des MANETs zu beeinträchtigen.

Auch die Verfügbarkeit eines Wireless LANs kann ein böswilliger Angreifer durch Störung des physikalischen Übertragungsmediums oder durch Ausnutzung von Lücken in den verwendeten Übertragungsprotokollen beeinträchtigen. Im Gegensatz zu MANETs sind die hierfür relevanten Protokolle jedoch nicht der Vermittlungs-, sondern der ISO/OSI-Schicht 2 zugeordnet. Bekannte Schwachstellen existieren dabei sowohl in dem ursprünglichen Standard IEEE 802.11 als auch in seinen Erweiterungen.

*Zugriff auf übertragene Informationen:* Beim Zugriff auf Informationen ist zwischen Inhalts- und Signalisierungsdaten zu unterscheiden. Die Gründe eines Zugriffs hierauf sind vielfältig und reichen von der Analyse und Veränderung von Anwendungsdaten bis hin zur Gewinnung von Informationen über einzelne Netzteilnehmer, etwa zur Bestimmung ihrer Identität oder ihres Aufenthaltsortes. Essentielle Angriffstechnik für diesen Zweck ist neben dem Abhören des Übertragungsmediums wiederum die Manipulation des Ad-hoc-Routings. Diesmal mit dem Ziel, Datenverkehr zum Angreifer umzulenken, wie dies beispielsweise bei einem sogenannten *Wurmloch*-Angriff der Fall ist. Je nach Ausprägung gefährdet der Zugriff auf die in einem MANET übermittelten Informationen somit die Integrität oder Vertraulichkeit.

Der Zugriff auf Signalisierungsdaten ist in Wireless LANs ebenso unkompliziert wie in MANETs, da diese grundsätzlich unverschlüsselt übertragen werden. Das Gleiche gilt für Inhaltsdaten, sofern der Access Point keine Verschlüsselung vorsieht. Aber selbst bei aktivierter Verschlüsselung existieren aktive und passive Angriffsmöglichkeiten, deren Erfolg jedoch von dem verwendeten Sicherheitsprotokoll abhängt. Nähere Details zu deren Schwachstellen finden sich in Kapitel 2.4.1 auf Seite 31.

Abbildung 2.4 zeigt einen Angriffsbaum in textueller Form, der mögliche Angriffsziele in mobilen Ad-hoc-Netzen näher erläutert. Eine detailliertere Analyse für das weit verbreitete DSR Protokoll, die sich problemlos für andere Ad-hoc-Routing-Protokolle adaptieren lässt, findet sich in [Kar03, Kapitel 6]. Abbildung 2.5 auf der nächsten Seite zeigt eine entsprechende Darstellung der häufigsten Angriffsziele für Wireless LANs.

**ANGRIFFSBAUM 1:** MANET angreifen
**OR** 1. Einsparen eigener Ressourcen
- **OR** 1.1 Sabotage der Wegewahl
  - **OR** 1.1.1 Keine Weiterleitung von Routing-Daten
  - 1.1.2 Routing-Daten/Topologie modifizieren
  - 1.1.3 Aus aktiver Route aussteigen
- 1.2 Sabotage der Paketweiterleitung

2. Beeinträchtigung der Funktionsfähigkeit des Netzes
- **OR** 2.1 Datenpakete zerstören
  - **OR** 2.1.1 Störung der Funkschnittstelle (Jamming)
  - 2.2 Überlasten von Komponenten
  - **OR** 2.2.1 Überlasten direkter Nachbarn
    - **OR** 2.2.1.1 Senden von falschen Paketen an direkte Nachbarn
    - 2.2.2 Überlasten beliebiger Knoten eines Netzes
    - **OR** 2.2.2.1 Versenden gefälschter Pakete an beliebige Knoten
  - 2.3 Korrekte Routing-Funktion stören
  - **OR** 2.3.1 Pakete im Netz verloren gehen lassen *(„Black-Hole-Routing“)*
  - 2.3.2 Topologiedaten zerstören

3. Zugriff auf übertragene Informationen
- **OR** 3.1 Durch den eigenen Knoten geleitete Datenpakete abgreifen
- 3.2 Datenpakete gezielt zum Knoten umleiten
- **OR** 3.2.1 Angreiferknoten geeignet positionieren
- 3.2.2 Routing manipulieren

Abbildung 2.4: Angriffsziele in mobilen Ad-hoc-Netzen

### 2.2.3 Fazit

Die Gefährdung von MANETs ist alleine aufgrund ihres offenen Charakters augenscheinlich. Vor allem der im Vergleich zu herkömmlichen Netzen neue Typus des egoistischen Angreifers stellt dabei eine große Bedrohung dar, da er der typischen Verhaltensweise eines ökonomisch agierenden Benutzers folgt. So ist es sinnvoll, jegliche freiwillige Mitarbeit an einem MANET zu verweigern, solange die bereitgestellte Dienstgüte den persönlichen Anforderungen noch entspricht. Ein rational agierender Knoten wird daher stets versuchen, diesen Trade-off möglichst genau zu bestimmen und entsprechend zu handeln. Die bisher in mobilen Ad-hoc-Netzen mehr oder weniger implizit unterstellte altruistische Natur der einzelnen Netzknoten erscheint hingegen im Allgemeinen wenig realistisch.

**ANGRIFFSBAUM 2:** WLAN angreifen
**OR** 1. Beeinträchtigung der Funktionsfähigkeit des Netzes
- **OR** 1.1 Störung der Funkschnittstelle
- 1.2 DoS-Angriffe auf EAP [MA02]
- 1.3 DoS-Angriffe auf WPA-Michael [EA04a]
- 1.4 DoS-Angriffe auf IEEE 802.11 [BS03]
- **OR** 1.4.1 Versenden gefälschter Deauth/Disassoc-Nachrichten
- 1.4.2 Power-Save-Angriff
- 1.4.3 Massenhaftes Versenden gefälschter Auth/Assoc-Nachrichten *(Auth/Assoc-Flooding)*
- 1.4.4 Versenden fehlerhafter Auth-Nachrichten
- 1.4.5 RTS/CTS-Angriff
- 1.4.6 Massenhaftes Versenden gefälschter Beacon-Nachrichten *(Beacon Flooding)*

2. Zugriff auf Informationen
- **OR** 2.1 Übertragungsmedium abhören (*Sniffing*)
- 2.2 Man-in-the-Middle-Angriff
- **AND** 2.2.1 MAC Spoofing/Impersonation
- 2.2.2 Senden gefälschter Management-Frames
- 2.3 WEP-Schlüssel brechen
- **OR** 2.3.1 Zufälliges Ausprobieren des Schlüssels *(Brute Force)*
- 2.3.2 Wörterbuchangriff
- 2.3.3 (Erweiterter) FMS-Angriff [FMS01, TWP07, Kle08]
- 2.3.4 KoreK-Angriff [Kor04b]
- 2.4 Schlüsselbitstrom mittels Klartext-/Schlüsseltextangriff gewinnen
- **OR** 2.4.1 Ausnutzen der Shared-Key-Authentikation [BGW01]
- 2.4.2 Arbaugh-Angriff (DHCP-Discover) [Arb01]
- 2.4.3 chopchop Angriff [Kor04a]
- 2.4.4 IP Redirection Angriff [BHL06]
- 2.4.5 Layer 2 Fragmentation Angriff [BHL06]

Abbildung 2.5: Angriffsziele in Wireless LANs

Ähnliches gilt für Wireless LANs. Bedenkt man etwa den Reiz eines kostenlosen Internet-Zugangs, so erscheint das geschilderte egoistische Verhalten auch hier nicht weit hergeholt. Der Übergang von egoistischem zu böswilligem Verhalten ist jedoch fließend, benutzen doch beide Angreifertypen dieselben Techniken. Letztlich bestimmt ausschließlich die Intention des Angreifers sein Verhalten.

Die konsequente Ausnutzung von Schwachstellen auf unterschiedlichen Ebenen des ISO/OSI-Modells lässt zudem darauf schließen, dass nur ein holistischer Ansatz in der Lage sein wird, nachhaltigen Schutz zu bieten.

## 2.3 Anforderungen an mobile Sicherheitsarchitekturen

Die dargestellten Bedrohungen und Angriffe machen deutlich, dass zum Schutz mobiler Umgebungen spezielle Sicherheitsmaßnahmen dringend erforderlich sind. Zu deren Gestaltung lassen sich einige grundsätzliche Überlegungen anstellen.

### 2.3.1 Grundsätzliche Ideen zum Schutz mobiler Umgebungen

Ganger und Nagle stellen in [GN01] einen Ansatz für Netzsicherheit vor, dessen Kernidee die Entwicklung selbstschützender Einheiten darstellt. In Analogie zu mittelalterlichen Festungstaktiken wird die Errichtung eines Sicherheitsperimeters um jede Einheit gefordert; jede Einheit soll sich selbst bestmöglich schützen. Die Unterteilung in Einheiten erfolgt dabei recht feingranular bis hin zur Ebene von Systemkomponenten, wie etwa Netzwerk- oder Grafikkarte, Prozessor und Speichereinheiten. Die Autoren illustrieren die Vorteile ihres Ansatzes anhand einiger Anwendungsbeispiele, wie etwa die direkte Integration von Firewall-Eigenschaften in eine Netzwerkkarte. Neben der eigentlichen Errichtung des Sicherheitsperimeters um eine Einheit, also die Entwicklung selbstschützender Einheiten, identifizieren sie als wesentliches Forschungsproblem zur Realisierung ihres Ansatzes die Verwaltung mehrerer solcher selbstschützenden Einheiten. Neben der Frage, wie die globale Sicherheit der übergeordneten Einheit durch die Kopplung einzelner selbstschützender Einheiten gewährleistet werden kann, seien hierbei vor allem dynamische Aspekte relevant: Wie wirkt sich der erkannte Angriff auf eine Einheit auf das Verhalten der anderen Einheiten aus? Angedacht wird hier die Einführung eines netzwerkweiten Sicherheitsstatus, der in einem solchen Fall erhöht wird, um etwa bei anderen Einheiten verschärfte Sicherheitsmaßnahmen auszulösen. Auch die Kooperation einzelner Einheiten bei der Erkennung von Angreifern wird bereits implizit angedacht.

Zwar stellt die Arbeit nur grundsätzliche Ideen vor und bleibt konkrete Umsetzungen schuldig, sie liefert dennoch gute Ansätze zur Sicherung mobiler Umgebungen. So löst etwa eine tiefgreifende Sicherheitsstrategie, jede einzelne Einheit bestmöglich zu schützen, auch das in mobilen Umgebungen bestehende Problem verschwimmender Netzgrenzen. Es erscheint daher vielversprechend, den Eigenschutz mobiler Endgeräte auszubauen.

Im Gegensatz zu den allgemeinen Überlegungen Gangers und Nagles diskutieren Beyah et al. [BCC06] bereits konkrete neue Sicherheitsbedrohungen, die durch die fortschreitende Verbreitung von Wireless LANs geschaffen werden. Sie argumentieren, dass im Gegensatz zu bisherigen Angriffen in traditionellen Netzen zukünftig ein verstärktes Augenmerk auf Insider-Angriffe und damit auf den gegenseitigen Schutz autorisierter Nutzer zu legen sei. Hieran änderten auch neue präventive Sicherheitsstandards wie etwa WEP oder IEEE 802.11i nichts, da diese lediglich den Status Quo kabelgebundener Netze wiederherstellen und somit drahtlose Netze ausschließlich vor externen Angrei-

fern schützen. Die Autoren fordern stattdessen eine Ergänzung solcher präventiver Sicherheitsmaßnahmen durch überwachende Maßnahmen, um Sicherheitsrichtlinien durchsetzen und legitime Nutzer voreinander schützen zu können. Das Gefährdungspotential von Insider-Angriffen belegen sie exemplarisch anhand eines Szenarios zur Verbreitung eines Wurms über verdeckte drahtlose Kanäle. Hierfür ziehen sie aus der Epidemologie bekannte Infektionsmodelle zur Modellierung der Ausbreitungsrate des Wurms heran. Bei der Suche nach möglichen Techniken zur Verminderung der Ausbreitung argumentieren sie, dass präventive Techniken wie etwa Personal Firewalls nicht ausreichend sind, da sich der Wurm hinter zulässigen Diensten maskieren kann. Stattdessen wird der Einsatz eines verteilten Intrusion Detection Systems vorgeschlagen.

Auch Beyah et al. bleiben eine konkrete Systemimplementierung schuldig, geben aber eine recht gute Einschätzung des Gefährdungspotentials durch Insider-Angriffe in mobilen Netzen. Problematisch ist hierbei vor allem die Tatsache, dass legitime Nutzer auch unwissentlich an einem Angriff beteiligt sein können (Stepping-Stone-Prinzip). Somit wird deutlich, warum präventive Maßnahmen, wie etwa der IEEE 802.11i Standard zwar zwingend notwendig, aber nicht ausreichend für eine umfassende Sicherheitslösung sind.

### 2.3.2 Funktionale Anforderungen

Im folgenden werden nun die angestellten Überlegungen zum Schutz mobiler Umgebungen weiter systematisiert. Hierzu werden Anforderungen an ein System zum Schutz mobiler Endgeräte ermittelt, wobei sich die vorliegende Arbeit auf den Schutz der mobilen Kommunikation konzentriert. Primäres Ziel ist zunächst das Ausschließen möglicher Bedrohungen (vgl. Kapitel 1.1 auf Seite 2).

**/R1/ Prävention**
Neben der Beeinträchtigung der Funktionalität werden der unbefugte Gewinn von Informationen oder deren unberechtigte Modifikation als die grundsätzlichen Bedrohungen für ein IT-System angesehen [VK83, Zen89]. Die Sicherheit der in drahtlosen Netzen übertragenen Daten lässt sich somit anhand folgender drei Schutzziele formulieren, die eine Sicherheitsarchitektur mittels geeigneter Sicherheitsmaßnahmen durchsetzen muss:

- Vertraulichkeit (engl. *confidentiality*): Alle übermittelten Nachrichteninhalte sind nur den beteiligten Kommunikationspartnern zugänglich.
- Integrität (engl. *integrity*): Die übermittelten Nachrichteninhalte sind korrekt, vollständig und aktuell oder dies ist erkennbar nicht der Fall.
- Verfügbarkeit (engl. *availability*): Das Netzwerk ermöglicht berechtigten Kommunikationspartnern auf Wunsch jederzeit den Austausch von Nachrichten.

Die hierzu korrespondierenden Bedrohungen (unbefugter Informationsgewinn, unbefugte Modifikation von Informationen, unbefugte Beeinträchtigung der Funktionalität) lassen sich darüber hinaus auf die Kommunikationsumstände anwenden [WP00]. Man spricht dann von den Schutzzielen Anonymität (engl. *anonymity*) bzw. Unbeobachtbarkeit (engl. *unobservability*), Zurechenbarkeit (engl. *accountability*) sowie Erreichbarkeit (engl. *reachability*) bzw. juristische Durchsetzbarkeit (engl. *legal enforcability*) . Bisher fehlen für deren Umsetzung jedoch speziell auf drahtlose Netze zugeschnittene Sicherheitsmechanismen, weshalb im weiteren Verlauf ausschließlich die Sicherheit der Kommunikationsinhalte betrachtet wird.

Der dynamische und prozessorientierte Charakter der IT-Sicherheit erfordert außerdem das Erkennen möglicher Zwischenfälle, die die andauernde Einhaltung der formulierten Sicherheitsziele beeinträchtigen können.

**/R2/ Überwachung**
Die Überwachung ermöglicht das Feststellen anomalen Systemverhaltens und damit die Erkennung von Schutzzielverletzungen. Die Gründe hierfür können vielschichtig sein. Sie reichen von unzulässigem Verhalten des Benutzers über falsch konfigurierte, fehlerhaft implementierte bis hin zu gänzlich fehlgeplanten Sicherheitsmaßnahmen. Darüber hinaus kann mittels Überwachung auch die Effektivität der eingesetzten Sicherheitsmechanismen abgeschätzt werden. Insbesondere stellt Überwachung einen möglichen Schutz vor Angriffen durch legitime Benutzer dar. Man spricht hierbei von einem *Insider-Angriff* bzw. einem *Innentäter*. Der Angriff selbst wird zwar nicht verhindert, seine Auswirkungen lassen sich jedoch im Nachhinein nachvollziehen und zumindest teilweise rückgängig machen.

Nach dem Erkennen einer Schutzzielverletzung muss schließlich dafür Sorge getragen werden, die Systemsicherheit auch weiterhin zu gewährleisten.

**/R3/ Reaktion**
In Folge einer erkannten Schutzzielverletzung müssen adäquate Gegenmaßnahmen eingeleitet werden, um die Systemsicherheit wieder herzustellen. Die Wahl der passenden Gegenmaßnahme hängt dabei von der Ursache und den Auswirkungen des beobachteten Zwischenfalls ab. Die möglichst genaue Erkennung einer Schutzzielverletzung ist also die Grundlage für eine effektive Reaktion.

Diese drei Anforderungen legen die grundlegende Funktionalität einer Sicherheitsarchitektur für mobile Endgeräte fest. Im folgenden Abschnitt werden darüber hinaus notwendige nichtfunktionale Anforderungen an eine solche Architektur formuliert.

### 2.3.3 Nichtfunktionale Anforderungen

Mobile Endgeräte und Netze, wie sie in Kapitel 2.1 vorgestellt wurden, weisen einige essentielle Unterschiede zu herkömmlich vernetzten Desktop-Computern auf.

*Geräteressourcen:* Die Leistungsfähigkeit mobiler Endgeräte ist verglichen mit stationären Computern nicht nur deutlich eingeschränkt, sondern variiert auch noch sehr stark zwischen den einzelnen Gerätekategorien. Dies gilt für sämtliche Ressourcen, wie etwa Rechenkapazität, flüchtigen und nichtflüchtigen Speicher und Datenrate der Netzverbindung aber auch für die Energieversorgung, sprich die Akkuleistung.

*Nutzerschnittstelle:* Neben der Leistungsfähigkeit variieren die einzelnen Gerätegattungen insbesondere hinsichtlich der Interaktionsmöglichkeiten mit dem Nutzer. So ähnelt die Arbeit mit mobilen Standardcomputern zwar sehr der mit stationären Geräten, alle anderen Gerätegattungen weisen aber deutliche Unterschiede hinsichtlich der Ein- und Ausgabeschnittstelle auf. Zwar hat sich die Darstellungsqualität mobiler Anzeigen in den letzten Jahren signifikant verbessert, die Größe der Anzeige bleibt jedoch durch die geringen Gehäuseabmessungen prinzipiell limitiert. Auch auf die bei Standardcomputern etablierten Eingabegeräte Maus und Tastatur kann im mobilen Umfeld nicht oder nur begrenzt zurückgegriffen werden. Ferner gibt es Geräte, die keine direkte Interaktion mit dem Nutzer vorsehen und daher über keinerlei direkte Nutzerschnittstellen verfügen. Ein Beispiel hierfür sind Chipkarten oder Bordcomputer, die nur über spezielle Lesegeräte bedient werden können.

*Transportierbarkeit:* Eine wesentliche Voraussetzung für Mobilität ist die Transportierbarkeit der verwendeten Endgeräte; diese müssen klein und leicht sein. Aufgrund ihrer kompakten Abmessungen lassen sich mobile Endgeräte jedoch wesentlich einfacher entwenden als ihre stationären Pendants, wodurch ein Angreifer Zugriff auf sicherheitsrelevante Informationen erlangen kann. In der Folge kann ein gestohlenes Gerät dann für einen Insider-Angriff missbraucht werden.

*Netzzugang:* Mobile Standardcomputer vereinen bereits seit längerem Schnittstellen zu praktisch allen Typen drahtloser lokaler Netze. Auch die bisher über extern angekoppelte Geräte realisierte Verbindung mit drahtlosen Fernnetzen findet mehr und mehr Einzug direkt in die Endgeräte. Erste Business-Notebooks verfügen bereits über eine UMTS-Schnittstelle. Bei der Unterstützung von WiMAX ist eine ähnliche Entwicklung absehbar. Die einzelnen Übertragungsstandards weisen dabei unterschiedliche Sicherheitsprobleme auf. Zu beachten sind ferner die Implikationen, die beim Wechsel zwischen verschiedenen Übertragungstechniken auftreten können.

*Gerätezentrierung:* Je nach verwendeter Netzzugangstechnik ist zudem die technische Verfügbarkeit von Infrastrukturmaßnahmen eingeschränkt. Ein Beispiel hierfür sind Ad-hoc-Netze, wie sie etwa im Standard IEEE 802.11 definiert werden. Dies führt unter anderem dazu, dass im Gegensatz zu herkömmlichen Netzen die

einzig wirklich gesicherte Perimetergrenze durch das Endgerät selbst gezogen wird. In mobilen Umgebungen kommt somit das Weakest-Link-Paradigma [Sch00] deutlich stärker zum Tragen, nach dem die Systemsicherheit als eine Kette einzelner Maßnahmen betrachtet wird, wobei das schwächste Glied in der Kette die Gesamtsicherheit bestimmt.

*Nutzerzentrierung:* Das Fehlen einer Infrastruktur lässt sich auch auf den personellen Bereich ausdehnen. So werden mobile Endgeräte viel stärker als persönliches Eigentum eines einzelnen Nutzers angesehen als dies bei einem herkömmlichen Personal Computer der Fall ist. Zusammen mit der Mobilität der Endgeräte führt dies dazu, dass diese in der Regel nicht mehr der Kontrolle durch einen Systemadministrator unterliegen. Vielmehr ist jeder Benutzer selbst für die Wartung und Pflege seiner persönlichen Geräte zuständig. Hier stellt sich die Frage, ob er dazu auch willens und in der Lage ist.

Aus diesen Beobachtungen lassen sich weitere Anforderungen ableiten, die bei der Entwicklung der angestrebten Sicherheitsarchitektur zu berücksichtigen sind.

**/R4/ Skalierbarkeit**
Infolge der beschränkten Geräteressourcen muss eine Sicherheitsarchitektur berücksichtigen, dass sie in Konkurrenz zu den eigentlichen funktionalen Eigenschaften eines mobilen Geräts steht und diese nicht über Gebühr einschränken darf. Es müssen deshalb Mittel bereitgestellt werden, die auch ressourcenschwachen Geräten ein Mindestmaß an Sicherheit ermöglichen ohne ihre eigentliche Funktionalität zu sehr zu beeinträchtigen.

**/R5/ Usability und Ergonomie**
Unter Gebrauchstauglichkeit (engl. *Usability*) wird die allgemeine Eignung eines IT-Systems verstanden, dem Benutzer im Rahmen eines festgelegten Benutzungskontexts das zufriedenstellende Erreichen zuvor definierter Ziele zu ermöglichen. Diese allgemeine Anforderung sollte prinzipiell von jedem IT-System erfüllt werden. Im mobilen Umfeld muss dabei ein besonderes Augenmerk auf die Interaktion mit dem Nutzer gelegt werden. Sie muss an die durch das mobile Endgerät vorgegebenen Gegebenheiten angepasst werden. Um etwa auch Geräte ohne direkte Nutzerschnittstelle zu berücksichtigen, sollte zumindest eine grundlegende Funktionalität der geplanten Lösung vollkommen autonom, also ohne jegliche Eingriffe durch den Benutzer, möglich sein.

Die explizite Nutzerzentrierung mobiler Geräte stellt zudem hohe Ansprüche an die Ergonomie der zu entwickelnden Sicherheitsarchitektur. Insbesondere müssen die Fähigkeiten und Aufgaben des individuellen Benutzers berücksichtigt werden. Die geplante Lösung darf diesen weder mit zu vielen Detailinformationen überfordern, noch ihn im Unklaren über eine potentiell gefährliche Situation lassen. Im Gegensatz zu den meisten herkömmlichen Sicherheitslösungen, die in der Regel fundiertes Expertenwissen voraussetzen, muss eine Sicherheitsarchitektur für mobile Endgeräte daher im besonderen Maße den jeweiligen Wissensstand des einzelnen Benutzers berücksichtigen.

**/R6/ Sicherheit**
Ein Sicherheitssystem stellt natürlich selbst ein Ziel für mögliche Angriffe dar und muss daher entsprechend geschützt werden. Zur Wahrung der technischen Sicherheit kann hierbei auf existierende präventive Techniken etwa aus den Bereichen der Kryptographie oder des Trusted Computing zurückgegriffen werden. Auf diese Weise kann die Sicherheit auf dem einzelnen Endgerät aber auch bei der Datenübertragung zwischen verschiedenen Endgeräten gewährleistet werden.

**/R7/ Verlässlichkeit**
Der Schutz der Datenübertragung zwischen kooperierenden Endgeräten ist zwar notwendig, alleine aber noch nicht ausreichend zur Wahrung der Sicherheit kooperativer Sicherheitslösungen in offenen Umgebungen. Im Gegensatz zu den implizit vorgegebenen Vertrauensbeziehungen infrastruktureller Lösungen fehlt den frei kooperierenden Endgeräten eine gemeinsame Vertrauensbasis. Beim Eingehen einer Kooperationsbeziehung muss daher ein verstärktes Augenmerk auf die Verlässlichkeit (engl. *Dependability*) des potentiellen Kooperationspartners gelegt werden. Mit anderen Worten, es muss nicht nur die systeminhärente Sicherheit, sondern auch die der Systemumgebung berücksichtigt werden.

Die im Fachbereich Sicherheit der Gesellschaft für Informatik e. V. aufgegangene Fachgruppe Verlässliche IT-Systeme (VIS) definierte die Verlässlichkeit von IT-Systemen dabei wie folgt [VIS]:

> *Die Verlässlichkeit eines informationstechnischen Systems (IT-Systems) ist ein Bewertungsmaßstab, den insbesondere die Benutzer und Betreiber an dieses System anlegen. Verlässlichkeit umfasst ein Bündel von Systemeigenschaften, die über die klassischen Sicherheitsanfordrungen der Verfügbarkeit, Integrität und Vertraulichkeit hinausgehen. Ein umfassendes Kriterium ist die Verantwortbarkeit der Nutzung eines IT-Systems [. . . ]*

Zur Bewertung der Verlässlichkeit wird daher die technische Interpretation der drei Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit in Anlehnung an [Ste08] um eine semantische Komponente erweitert. Die Informationstheorie versteht unter der Semantik einer Information die Bedeutung dieser Information. Demnach kann die semantische Erfüllung der drei Sicherheitsziele wie folgt definiert werden:

- *Semantische Vertraulichkeit* oder *Diskretion* ist gegeben, wenn keiner der an einer Nachrichtenübermittlung beteiligten Kommunikationspartner Teile der übermittelten Nachrichten Dritten auf irgendeine Weise zugänglich macht.
- *Semantische Integrität* oder *Konformität* ist erfüllt, wenn die Kommunikationspartner inhaltlich korrekte, vollständige und aktuelle Nachrichten übermitteln. Um sich subjektiv betrachtet konform zu verhalten, muss ein Kommunikationspartner die

in ihn gesetzten Erwartungen erfüllen. Er darf insbesondere nicht wissentlich Fehlinformationen verbreiten.

- Von *semantischer Verfügbarkeit* oder *Kooperationsbereitschaft* spricht man, wenn potentielle Partner dem Wunsch nach Zusammenarbeit nachkommen und zum Austausch von Nachrichten bereit sind.

Sowohl die Diskretion als auch die Kooperationsbereitschaft kann in aller Regel objektiv beurteilt werden. Die Feststellung von Konformität hingegen gestaltet sich meist schwieriger, da die Einschätzung konformen Verhaltens sowohl vom Standpunkt des Betrachters als auch von weiteren Faktoren, wie etwa dem Zeitpunkt der Beurteilung, abhängt.

**/R8/ Universalität**
Um die Vielzahl möglicher Übertragungstechniken zu unterstützen, muss eine Sicherheitsarchitektur einerseits von der genutzten Netzzugangstechnik abstrahieren, andererseits aber auch auf deren spezifische Merkmale eingehen. Ein holistischer Ansatz erfordert insbesondere die Einbeziehung unterschiedlichster Sicherheitsmechanismen, die sich auf beliebige Schichten des OSI-Referenzmodells erstrecken können.

**/R9/ Dezentralisierung**
Eine Sicherheitsarchitektur für mobile Endgeräte muss das mögliche Fehlen infrastruktureller Maßnahmen berücksichtigen. Sie muss insbesondere den fehlenden Zugang zu zentralen Sicherungsmaßnahmen kompensieren, indem sie den Schutz des einzelnen Endgeräts in den Vordergrund stellt, gleichzeitig aber die Sicht auf das System als Ganzes nicht verliert.

In Abbildung 2.6 auf der nächsten Seite werden die genannten Anforderungen und ihre Einflussfaktoren zusammengefasst. Die wesentlichen Abhängigkeiten der einzelnen Anforderungen von den Einflussfaktoren werden durch Pfeile dargestellt.

Farbig hervorgehoben sind hierbei die Aspekte Überwachung, Skalierbarkeit, Verlässlichkeit, Universalität und Dezentralisierung, die den Schwerpunkt dieser Arbeit bilden.

Nachdem nun sowohl die Einsatzumgebung der geplanten Sicherheitsarchitektur vorgestellt als auch die von ihr zu erfüllenden Anforderungen festgelegt wurden, sollen im Folgenden zunächst in der Literatur beschriebene Sicherheitsarchitekturen für mobile Umgebungen erläutert werden. Allgemeine Vorüberlegungen wurden bereits in Kapitel 2.3.1 dargestellt. Nicht zuletzt der große Erfolg des IEEE 802.11 Standards hat jedoch zu regen Forschungsaktivitäten geführt, die sich in einer Vielzahl von Publikationen und konkreten Systementwürfen niederschlagen.

Die folgende Darstellung unterscheidet diese zunächst hinsichtlich ihrer grundsätzlichen funktionellen Beschaffenheit nach präventiven, reaktiven und angriffstoleranten

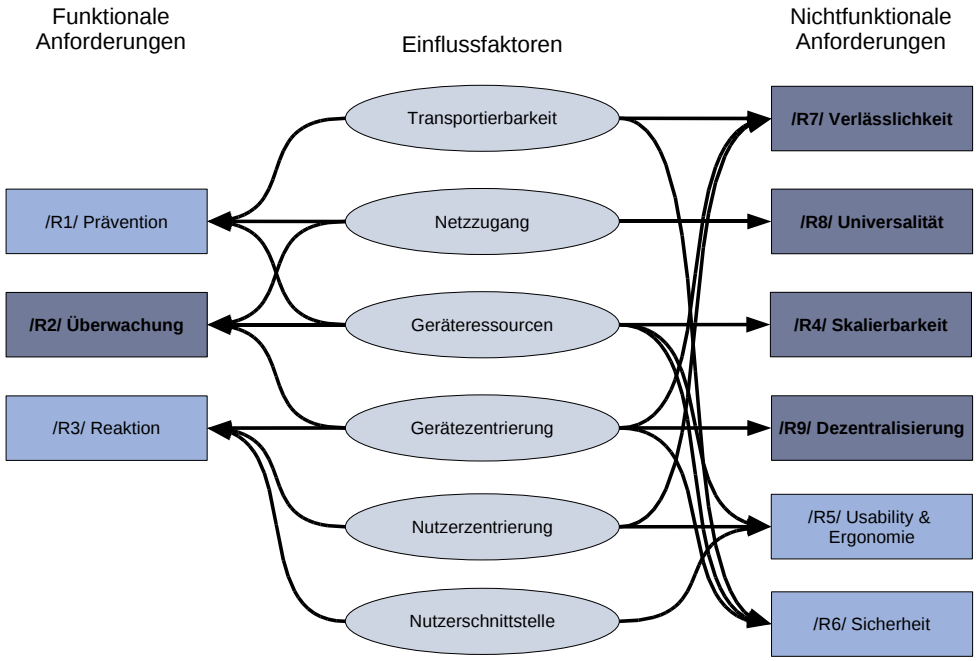



Abbildung 2.6: Anforderungen an eine Sicherheitsarchitektur für mobile Endgeräte

Architekturen. Darüber hinaus wird jeweils zwischen Ansätzen für mobile Ad-hoc-Netze und solchen für mobile infrastrukturelle Netze unterschieden. Es fällt auf, dass erstgenannte in der Regel auf Vermittlungsebene gemäß des ISO/OSI-Referenzmodells angesiedelt sind, während die Lösungen für infrastrukturelle Netze auf der Sicherungsschicht ansetzen. Dies deckt sich jedoch mit den Betrachtungen in Kapitel 2.2, wonach Angriffe in Ad-hoc-Netzen auf die Wegewahl und Paketweiterleitung durch kooperierende Knoten eines Ad-hoc-Netzes, solche auf herkömmliche Wireless LANs hingegen eher auf die einzelne Verbindung eines mobilen Endgeräts zum Access Point zielen.

Die betrachteten Systeme werden schließlich dahingehend untersucht, ob und inwieweit sie die in diesem Abschnitt aufgestellten Anforderungen erfüllen. Ziel ist es, bisher unzureichend behandelte Aspekte zu identifizieren und damit den Rahmen für die weitere Arbeit abzustecken.

## 2.4 Präventive mobile Sicherheitsarchitekturen

Präventive Sicherheitsarchitekturen konzentrieren sich auf die Umsetzung der funktionalen Anforderung */R1/ Prävention* und damit auf die a priori Verhinderung der Kompromittierung von Schutzzielen. Sie hängen wesentlich von der zugrunde liegenden Übertragungstechnik ab und werden daher meist als Teil von deren Spezifikation definiert.

Im folgenden werden zunächst typische präventive Sicherheitsmaßnahmen am Beispiel von Schutzmechanismen für die Sicherungsschicht von Wireless LANs kurz vorgestellt. Weitergehende Einzelheiten hierzu können der einschlägigen Literatur und den jeweiligen Standards entnommen werden.

Vergleichbare Schutzmaßnahmen wurden sowohl für andere Übertragungstechniken wie auch auf anderen Ebenen des Referenzmodells festgelegt. Exemplarisch seien hier die zahlreichen Arbeiten zum sicheren Routing in mobilen Ad-hoc-Netzen genannt, die im Folgenden ebenfalls übersichtsartig dargestellt werden.

### 2.4.1 Schutzmaßnahmen auf Sicherungsebene

Für drahtlose Netze nach IEEE 802.11 existieren mit WEP, WPA und RSN derzeit drei weit verbreitete Standards zum Schutz der Sicherungsschicht. Neben der Gewährleistung von Vertraulichkeit und Integrität der Kommunikation zwischen mobilen Endgeräten und Access Points bieten sie auch Mechanismen zur Authentifizierung und Zugriffskontrolle, wobei sie nicht nur funktionell, sondern auch qualitativ gravierende Unterschiede aufweisen.

Im folgenden werden die einzelnen Verfahren und ihre Unterschiede kurz vorgestellt. Für eine tiefergehende Betrachtung sei beispielsweise auf [EA04b] verwiesen.

#### Wired Equivalent Privacy (WEP)

Wie der Name schon andeutet, war das erklärte Ziel von WEP beim Betrieb eines WLANs zumindest das Sicherheitsniveau eines drahtgebundenen Netzes wiederherzustellen. Hierzu sollte die Vertraulichkeit und Integrität der Datenübermittlung gewährleistet und die Teilnahme am Datenaustausch durch Authentifikation und Zugangskontrolle reglementiert werden. Mittlerweile gilt WEP jedoch als vollständig gebrochen [BHL06]. Daher werden im Folgenden die verwendeten Verfahren und ihre Schwachstellen nur kurz erläutert.

**WEP-Vertraulichkeit** Zur Wahrung der Vertraulichkeit verwendet WEP einen zwischen den Beteiligten vorab manuell ausgetauschten geheimen WEP-Schlüssel von 40 oder 104 Bit Länge. Zur Verschlüsselung wird die Stromchiffre RC4 verwendet, die mit der Eingabe des Schlüssels einen Strom von Pseudozufallszahlen erzeugt, der mittels

XOR mit dem Klartext verknüpft wird. Stromchiffren sind jedoch insbesondere gegen Known-Plaintext-Angriffe anfällig. Daher wird der statische Schlüssel für jedes Datenpaket um einen individuell gewählten 24-Bit Initialisierungsvektor (IV) ergänzt. Der gemeinsame geheime WEP-Schlüssel und der IV bilden damit den RC4-Schlüssel. Damit der Empfänger diesen Schlüssel rekonstruieren kann, wird der IV zusammen mit dem Datenpaket im Klartext übermittelt.

Hauptkritikpunkte an diesem Verfahren sind die zu kurzen Schlüssellängen und der zu kurze IV. Fluhrer, Mantin und Shamir demonstrierten in ihrem Aufsatz von 2001 zudem die Existenz sogenannter schwacher RC4-Schlüssel mit denen sich der WEP-Schlüssel rekonstruieren lässt [FMS01]. Der Angriff wurde bald darauf in frei verfügbaren Werkzeugen implementiert und immer weiter verfeinert, so dass der WEP-Schlüssel heute binnen kürzester Zeit ermittelt werden kann [TWP07].

**WEP-Authentifikation** WEP legt zwei Authentifikationsschemata fest. Bei der *Open System Authentication* finden keinerlei Überprüfungen statt, jeder erhält Zugang zum WLAN. *(Pre)Shared Key (PSK) Authentication* verwendet hingegen den geheimen WEP-Schlüssel und ein Challenge-Response-Protokoll, um den Zugriff illegitimer Nutzer auf das WLAN zu unterbinden. Hierbei sendet der Access Point eine 1024 Bit lange Zufallszahl an den Nutzer, der diese RC4-verschlüsselt an ihn zurücksendet, um so die Kenntnis des geheimen WEP-Schlüssels und damit seine Berechtigung für den Zugriff auf das WLAN nachzuweisen.

Da der verwendete WEP-Schlüssel für alle Nutzer identisch ist, ist es nicht möglich, einzelne Benutzer individuell zu authentifizieren. Zusammen mit der äußerst mühsamen Verteilung des Schlüssels per Hand führt dies zu der außerordentlich schlechten Skalierbarkeit von WEP. Darüber hinaus erfolgt keine Authentifikation des Access Points gegenüber dem Nutzer.

Gravierender ist jedoch die Tatsache, dass ein Angreifer durch Abhören der Challenge-Response-Authentifikation ein Klartext-/Schlüsseltextpaar erhält. Unter Ausnutzung der Eigenschaften des verwendeten RC4-Algorithmus, kann er sich damit in Zukunft selber authentifizieren, indem er mit der XOR-Verknüpfung aus Challenge und Response zu jeder neuen Challenge die passende Response berechnet. Dadurch wird auch die Integrität gefährdet, da der so berechnete Bitstrom zur Einschleusung gefälschter Pakete genutzt werden kann. Von der Verwendung der Shared Key Authentification wird deshalb prinzipiell abgeraten [DGK+03]. In der Praxis wird somit keinerlei Authentifizierung durchgeführt.

**WEP-Zugriffskontrolle** WEP sieht keine wirkliche Zugriffskontrolle vor. Jedoch erlauben die meisten Access Points die Einschränkung der berechtigten Nutzer, indem sie eine Liste zulässiger MAC-Adressen von Netzwerkkarten verwalten. Dieser rudimentäre Schutz ist jedoch nicht besonders wirkungsvoll, da zulässige MAC-Adressen durch

Abhören des normalen Netzverkehrs problemlos ermittelt und anschließend gefälscht werden können.

**WEP-Integrität** Bei der Übertragung von Nachrichten wird mittels des CRC-32-Algorithmus eine Prüfsumme berechnet und gemeinsam mit den Nutzdaten des Datenpakets übertragen. Das CRC-Verfahren stellt zwar eine effiziente Möglichkeit zur Erkennung von Bitübertragungsfehlern dar, jedoch handelt es sich hierbei nicht um eine kryptographisch sichere Hashfunktion. Es ist somit in keinster Weise zur Gewährleistung der Integrität einer Nachricht geeignet. Tatsächlich ist es relativ einfach möglich, durch gezielte Modifikationen einen Datenstrom zu erzeugen, der den gleichen CRC-Wert wie eine gegebene Nachricht hat. Neben der viel zu kurzen Prüfsumme von nur 32 Bit, ist dies vor allem in der Linearität des CRC begründet. Dadurch lässt sich aus der Menge der veränderten Bits der Originalnachricht problemlos die Menge der anzupassenden Bits berechnen, um den CRC-Wert an die geänderte Nachricht zu adaptieren.

Darüber hinaus wurde die Sicherstellung der Aktualität übertragener Nachrichten bei der Entwicklung von WEP völlig außer Acht gelassen. Mit anderen Worten, der Schutz vor Replay-Angriffen in WEP ist nicht mangelhaft, sondern schlicht und ergreifend inexistent [EA04b].

#### Wi-Fi Protected Access (WPA)

Nachdem sich die Unzulänglichkeiten von WEP immer deutlicher abzeichneten, begann die IEEE 802.11 Task Group mit der Verbesserung des WEP-Standards, die schließlich in der Definition einer vollständig neuen Sicherheitsarchitektur durch die *Task Group i* enden sollte. Bis zur Fertigstellung des Standards IEEE 802.11i benötigte die Industrie jedoch eine Interimslösung, um die bereits vorhandenen WLAN-Installationen besser zu schützen. Die Wi-Fi Alliance[1], ein Zusammenschluss namhafter Firmen aus dem Bereich Wireless LAN, legte daher auf Basis der bereits vorhandenen Arbeiten der Task Group i den sogenannten Wi-Fi Protected Access (WPA) Standard fest.

Ziel von WPA war die Beseitigung der dringlichsten Schwächen von WEP. So führt WPA eine neue Methode namens *Michael* ein, mit der übermittelte Nachrichten mittels sogenannter *Message Integrity Codes (MIC)* vor Manipulation geschützt werden. Die Verwendung von IEEE 802.1X in Kombination mit dem Extensible Authentication Protocol (EAP) verhindert effizient den Zugriff Unbefugter auf das WLAN. Noch immer aber wird RC4 als Verschlüsselungsalgorithmus verwendet, um Abwärtskompatibilität zu bereits vorhandenen Geräten zu ermöglichen. Zur Sicherstellung der Vertraulichkeit wurde daher das Temporal Key Integrity Protocol (TKIP) definiert. Namengebend war ein Verfahren zur Generierung temporärer RC4-Schlüssel, die sich regelmäßig ändern.

[1] http://www.wi-fi.org/ (Abruf: August 2008)

Insbesondere wird die Generierung schwacher Schlüssel verhindert. Darüber hinaus legt TKIP größere Schlüssel- und IV-Längen fest: 128 respektive 48 Bit.

Prinzipiell löst WPA die Probleme von WEP, ohne jedoch die grundlegend bemängelten Teile wie etwa den RC4-Algorithmus zu ersetzen. Trotzdem gilt WPA nicht zuletzt wegen seiner im Gegensatz zu WEP unter Beteiligung der kryptographischen Gemeinschaft durchgeführten Entstehung allgemein als sicher, selbst wenn für *Michael* bereits erste Schwächen bekannt geworden sind (siehe beispielsweise [EA04a]).

#### Robust Security Network (RSN)

Das im 2004 verabschiedeten Standard IEEE 802.11i vorgestellte RSN, auch bekannt unter dem Namen WPA2, löst die Probleme von WEP grundlegend. Dabei teilen sich RSN und WPA in weiten Teilen eine gemeinsame Architektur. Da RSN aber auf neue Hardware aufsetzt und nicht mehr zwingend Abwärtskompatibilität zu vorhandenen Systemen fordert, bietet es eine Vielzahl weiterer Verbesserungen. Zur Unterstützung vorhandener Altgeräte definiert IEEE 802.11i sogenannte *Transitional Security Networks (TSN)*, in denen sowohl RSN- als auch WEP-basierte Geräte parallel betrieben werden können.

Zu den hervorstechendsten Neuerungen gehört das *Counter-Mode/CBC-MAC Protocol (CCMP)*, ein von Grund auf neu entwickeltes Protokoll zur Gewährleistung der Vertraulichkeit und Integrität sowie zur Authentifizierung. CCMP stellt eine radikale Abkehr von der bisher verwendeten Stromchiffre RC4 dar. Statt dieser wird eine auf dem *Advanced Encryption Standard (AES)* basierende Blockchiffre eingesetzt. Die verwendete Schlüssellänge von 128 Bit gilt als sicher. Als Betriebsart für die Blockchiffre wird *Cipher Block Chaining (CBC)* verwendet, womit auch die Integritätssicherung erfolgt (*CBC-MAC*). Hierdurch werden die bei *Michael* möglichen Denial-of-Service-Angriffe unterbunden. CCMP wird daher allgemein als schwerer angreifbar angesehen als TKIP, wobei dessen weitere Verwendung optional möglich ist.

Genau wie bereits WPA ermöglicht RSN nicht nur die Authentifizierung über *Preshared Keys*, sondern auch mittels IEEE 802.1X und EAP. In diesem *Enterprise Mode* erfolgt die Schlüsselverteilung automatisch, wodurch WPA und RSN deutlich besser skalieren als WEP.

Der IEEE 802.11i Standard stellt sicherlich einen Meilenstein zum Schutz von Wireless LANs dar und gilt derzeit als sicher. Er schützt jedoch nicht vor Angriffen durch andere autorisierte Nutzer, was gerade im Hinblick auf die bereits erwähnte einfache Transportierbarkeit mobiler Geräte problematisch erscheint.

Tabelle 2.1 auf der nächsten Seite fasst die wesentlichen Unterschiede von WEP, WPA und RSN zusammen.

| | WEP | WPA | RSN |
|---|---|---|---|
| Vertraulichkeit | RC4 | RC4 | AES |
| Authentifikation | keine/PSK | PSK/EAP | PSK/EAP |
| Integrität | CRC-32 | MIC | CCM |
| Schlüssel | | | |
| – Typ | statisch | dynamisch | dynamisch |
| – Generierung | manuell | automatisch | automatisch |
| – Verteilung | manuell | automatisch (Enterprise Mode) | automatisch (Enterprise Mode) |
| – Länge | 40/104 Bit | 128 Bit | 128 Bit |
| Länge IV | 24 Bit | 48 Bit | 48 Bit |

Tabelle 2.1: Vergleich von WEP, WPA und RSN

### 2.4.2 Schutzmaßnahmen auf Vermittlungsebene

Während in infrastrukturellen Wireless LANs die Sicherheitsmaßnahmen auf Schicht 2 des ISO/OSI-Referenzmodells ansetzen, konzentrieren sich die Arbeiten in Ad-hoc-Netzen auf den Schutz des eingesetzten Verfahrens zur Wegewahl (engl. *Routing*). Die hierfür vorgeschlagenen Maßnahmen sind damit der Vermittlungsschicht zuzuordnen.

Hierbei ist es wichtig, das eigentliche Routing, also die Bestimmung eines Weges zwischen zwei beliebigen Knoten eines Netzes, strikt von der Nutzung dieses Weges im Rahmen der Weiterleitung (engl. *Forwarding*) von Nachrichten abzugrenzen. Es ist offensichtlich, dass ein Knoten durch präventive Maßnahmen nicht gezwungen werden kann, seine vorab während des Routing-Prozesses eingegangenen Verpflichtungen einzuhalten. Er ist im Gegenteil jederzeit frei in seiner Entscheidung, ob und wie er eine Nachricht weiterleitet.

Im folgenden wird daher zunächst eine allgemeine Einführung in die Methodik der Wegewahl in Ad-hoc-Netzen gegeben, bevor genauer auf deren Sicherheit eingegangen wird. Maßnahmen zum Schutz der Weiterleitung werden hingegen erst im Rahmen reaktiver und angriffstoleranter Sicherheitsarchitekturen in den Kapiteln 2.5 und 2.6 ab Seite 39 respektive 52 diskutiert.

#### Routing in mobilen Ad-hoc-Netzen

Die Wegefindung in paketorientierten Netzen erfüllt im wesentlichen zwei Aufgaben: das Auffinden von Wegen zwischen zwei beliebigen Knoten (*Route Recovery*) und die Anpassung auf mögliche Veränderungen der Netztopologie (*Route Maintenance*). Die hierzu vorgeschlagenen Verfahren können nach Art und Weise der dabei ausgetauschten Routing-Informationen grundsätzlich in zwei Gruppen unterteilt werden. Bei *Distance-*

*Vector-Verfahren* ermittelt jeder Knoten die Längen aller denkbaren Wege zu einem beliebigen Zielknoten. Datenpakete für einen bestimmten Zielknoten werden dann an denjenigen Nachbar weitergeleitet, der am Beginn des Pfades mit der kürzesten Gesamtdistanz steht. Alle Knoten tauschen ferner ihre individuellen Distanzvektoren mit ihren Nachbarn aus, so dass sich jeder nach und nach ein Bild des gesamten Netzes machen kann. *Link-State-Verfahren* verfolgen einen entgegengesetzten Ansatz. Anstatt Informationen zu beliebigen Knoten zu sammeln ermittelt jeder Knoten lediglich die Entfernungen zu seinen direkten Nachbarn und verbreitet diese Information an alle Knoten eines Netzes. Zum Preis eines höheren Ressourcenbedarfs ermöglichen Link-State-Verfahren damit ein besseres Gesamtbild eines Netzes.

Aufgrund der spezifischen Eigenschaften mobiler Ad-hoc-Netze lassen sich übliche Distance-Vector- und Link-State-Routing-Protokolle nicht für diese verwenden. Stattdessen wurden spezielle Routing-Verfahren entwickelt, die in drei Typen gegliedert werden. Sie unterscheiden sich dabei in der zentralen Frage, ob der Knoten eines Ad-hoc-Netzes Informationen zu allen möglichen Zielen oder nur zu solchen von direktem Interesse verwalten soll.

**Vorausplanendes Routing** Bei vorausplanenden Verfahren, auch als *proaktiv, global* oder *tabellenbasiert* bezeichnet, ermittelt ein Knoten vor Beginn der eigentlichen Kommunikation die Wege zu allen denkbaren Zielen innerhalb eines (Teil-)Netzes, selbst wenn er einzelne Pfade niemals benötigt. Die so generierten Routing-Tabellen müssen periodisch aktualisiert werden, indem ein Knoten sie beispielsweise in bestimmten Intervallen mit seinen direkten Nachbarn synchronisiert.

Für kleine Netze bis etwa hundert Knoten stellen proaktive Verfahren eine effiziente Lösung dar [AWD04, Tabelle 7]. Die vorausplanende Berechnung der Routen ermöglicht einen schnellen Kommunikationsaufbau aus Sicht der Applikation. Mit wachsender Anzahl der Knoten werden diese Verfahren jedoch immer schwerfälliger. So ist sowohl die Speicherung als auch die Aktualisierung der Routing-Informationen mit einem signifikanten Ressourcenbedarf verbunden.

**Bedarfsorientiertes Routing** Bedarfsorientierte oder auch *reaktiv* genannte Verfahren berechnen dagegen die Route von einem Quell- zu einem Zielknoten erst dann, wenn der Quellknoten tatsächlich eine Kommunikation mit dem Zielknoten initiiert, indem er eine *Route-Discovery-Funktion* ausführt. Für die Dauer ihrer Nutzung wird die so etablierte Route mit einer *Route-Maintenance-Prozedur* aufrecht gehalten. Sowohl Route-Discovery als auch Route-Maintenance erfordert dabei die Versendung von Kontrollnachrichten.

Reaktive Verfahren benötigen in der Regel deutlich weniger Bandbreite für die Signalisierung und sind damit auch für Netze mit mehreren hundert Knoten geeignet [AWD04, Tabelle 7]. Mit wachsender Größe des Netzes steigt allerdings auch der initiale

Aufwand zur Bestimmung einer Route, was bei vielen Anwendungen zu drastisch höheren Latenzzeiten führt.

**Hybride Routing-Protokolle** Um eine bessere Skalierbarkeit zu erreichen, kombinieren hybride Verfahren die beiden erstgenannten Ansätze, indem sie das Ad-hoc-Netz strukturieren und etwa für nahe beieinanderliegende Knoten proaktives, zwischen entfernteren Knoten dagegen reaktives Routing verwenden.

Hierdurch reduziert sich nicht nur der für die Signalisierung notwendige Aufwand, sondern es werden auch einzelne Fehlerstellen (engl. *Single Points of Failures*) und Flaschenhälse im Netz vermieden. Hybride Verfahren eignen sich daher auch für größere Ad-hoc-Netze mit bis zu tausend und mehr Knoten [AWD04, Tabelle 7].

Das Routing in mobilen Ad-hoc-Netzen hält eine Vielzahl von Herausforderungen bereit, die in der Vergangenheit Gegenstand ausführlicher Forschung waren. Dies führte zur Entwicklung unterschiedlichster Routing-Protokolle für MANETs. Für eine detaillierte vergleichende Übersichtsdarstellung der vorgeschlagenen Lösungen sei an dieser Stelle auf [AWD04, Per01] und [RT99] verwiesen.

Obwohl Sicherheit bereits frühzeitig [CM99] als eine wesentliche Anforderung für den praktischen Einsatz von MANETs identifiziert wurde, wird dieser Aspekt von den ursprünglichen Routing-Protokollen fast vollständig vernachlässigt [Per01]. Vielmehr wird für die beteiligten Knoten meist ein gutwilliges und fehlerfreies Verhalten angenommen, was wenig realistisch erscheint. In der Folge sind daher verstärkt Anstrengungen zur Entwicklung sicherer und robuster Ad-hoc-Routing-Protokolle unternommen worden, die im nächsten Abschnitt kurz vorgestellt werden. Für eine tiefergehende Betrachtung der Thematik sei bereits hier auf [AO05] und [HP04] verwiesen.

#### Sichere Wegewahl in mobilen Ad-hoc-Netzen

Ein sicheres und robustes Ad-hoc-Routing-Protokoll muss im wesentlichen zwei Eigenschaften erfüllen (vergleiche hierzu auch [Kar03, ZA02]):

- *Integrität* – Signalisierungsnachrichten können nur im Rahmen der korrekten Protokollfunktionalität verändert werden. Die Integrität umfasst insbesondere auch die *Aktualität* und *Authentizität* einer Routing-Nachricht.
- *Autorisation* – Es werden nur solche Signalisierungsnachrichten verarbeitet, die von dazu autorisierten Knoten im Rahmen der korrekten Protokollfunktionalität versendet wurden.

Darüber hinaus kann man auch die *Vertraulichkeit* der Routing-Nachrichten fordern, etwa um Außenstehenden einen Einblick in die Topologie des Ad-hoc-Netzes zu erschweren. Alle vorgeschlagenen Lösungen wandeln jedoch auf einem schmalen Grat zwischen

Sicherheit und Effizienz des Routing-Verfahrens, weshalb sie auf die Realisierung dieser Eigenschaft verzichten (vergleiche hierzu auch [HPT97]).

Ähnlich verhält es sich mit der *Verbindlichkeit* von Routing-Nachrichten, auch wenn diese beispielsweise hilfreich ist, um nicht konforme Knoten aus einem Ad-hoc-Netz auszuschließen [ZH99].

Quasi alle vorgeschlagenen Verfahren für den präventiven Schutz der Wegewahl in MANETs erweitern bereits existierende Ad-hoc-Routing-Protokolle um kryptographische Mechanismen, um die geforderten Eigenschaften sicherzustellen. Damit kann zwischen asymmetrischen, symmetrischen und hybriden Ansätzen unterschieden werden.

**Asymmetrische kryptographische Verfahren** Die Sicherstellung der Integrität von Routing-Nachrichten lässt sich beispielsweise durch die Verwendung digitaler Signaturverfahren realisieren. Für den sicheren Austausch der hierbei benötigten öffentlichen Schlüssel setzen solche asymmetrischen Verfahren eine vertrauenswürdige dritte Partei voraus. Je nach Verfahren muss diese Zertifizierungsinstanz permanent oder nur zeitweise für die Verifikation oder den Rückruf eines Zertifikats erreichbar sein, was je nach Anwendungsszenario ein Problem darstellen kann. Das Wiedereinspielen von Routing-Nachrichten wird durch die Ergänzung von Zeitstempeln und Nonces verhindert.

Die alleinige Verwendung asymmetrischer Verfahren erscheint aus mehreren Gründen problematisch. So stellt die ihr zugrundeliegende Voraussetzung der Existenz einer zentralen Zertifizierungsinstanz in der Praxis ein Problem dar, insbesondere wenn diese permanent erreichbar sein muss. Auch ist der Aufwand für die Operationen asymmetrischer kryptographischer Algorithmen im Hinblick auf die knappen Ressourcen mobiler Endgeräte sehr kritisch zu bewerten. Folglich werden in der Literatur mit Ausnahme des ARAN-Protokolls [SDL+02] praktisch keine rein asymmetrischen Ansätze diskutiert.

**Symmetrische kryptographische Verfahren** *Kryptographische Hash-Funktionen* sind ein probates Mittel zur Erkennung während der Übermittlung veränderter Nachrichten, die mit einem symmetrischen Verschlüsselungsverfahren zu einem sogenannten *Message Authentication Code (HMAC)* kombiniert auch zur Authentizitätssicherung verwendet werden können (vgl. beispielsweise [Sch96]). Zur effizienten Verteilung der hierzu notwendigen Schlüssel greifen viele Protokolle auf die erstmals in [Lam81] beschriebenen Hash-Ketten zurück. Eine *Hash-Kette* ist die wiederholte Anwendung einer Hash-Funktion $H$ auf einen zufällig gewählten Startwert $s$, woraus eine verkettete Folge von Hash-Werten $H_1, H_2, \ldots, H_n$ mit $H_1 = H(s)$ und $H_i = H(H_{i-1})$ resultiert. In umgekehrter Reihenfolge lässt sich diese zur Authentisierung von Nachrichten verwenden. So genügt die Preisgabe von $H_{i-1}$ bei Verwendung eines Message Authentication Codes mit dem Schlüssel $H_i$, um die Authentizität einer Nachricht zu beweisen.

Im Gegensatz zu asymmetrischen Verfahren sind symmetrische Verfahren deutlich effizienter und damit besser für ressourcenknappe mobile Endgeräte geeignet. Problematisch ist jedoch die Frage des Schlüsselaustauschs. Hierbei setzen bekannte Verfahren, wie etwa SRP [PH02], SEAD [HJP02] oder Ariadne [HPJ05], die Existenz paarweiser Sicherheitsbeziehungen zwischen den einzelnen Knoten oder eine synchrone Zeitbasis voraus. Beides erscheint in der Praxis unrealistisch.

**Hybride kryptographische Verfahren** Hybride Systeme, wie SAODV [Zap02, Zap06], SLSP [PH03] oder SDSR [Kar03], kombinieren asymmetrische und symmetrische kryptographische Verfahren, um einerseits die Performanz zu steigern, andererseits aber die Probleme bei der Initiierung multilateraler Sicherheitsbeziehungen zu mindern. Typischerweise werden dazu die unveränderlichen Teile einer Routing-Nachricht durch eine digitale Signatur vor Veränderungen geschützt. Der variable Anteil der Nachricht, also jener, der von den Knoten im Verlauf der Wegefindung verändert wird, ist hiervon ausgenommen. Er wird stattdessen mit effizienteren symmetrischen Verfahren gesichert.

Einerseits verbessert die geschickte Kombination asymmetrischer und symmetrischer Verfahren die Skalierbarkeit des Routing-Protokolls. Auf der anderen Seite erben die hybriden Ansätze zumindest teilweise auch deren Probleme. Es bedarf also auch hier einer genauen Prüfung und individuellen Entscheidung ob des Einsatzes eines solchen Protokolls für ein konkretes Anwendungsszenario.

#### Zusammenfassung

Die Frage der Sicherheit wurde bei der Wegewahl in mobilen Ad-hoc-Netzen lange Zeit ignoriert. Mittlerweile gibt es jedoch eine Vielzahl von Arbeiten, die zumindest einzelne Aspekte dieses Problems ansprechen. Leider werden dabei meist völlig unrealistische Grundannahmen getroffen, etwa zur Etablierung von Sicherheitsbeziehungen zwischen mobilen Knoten. Beispielsweise stoßen traditionelle Schlüsselmanagementlösungen aufgrund der nichthierarchischen Struktur mobiler Umgebungen schnell an ihre Grenzen. Zwar gibt es auch hierfür erste Lösungsvorschläge [MDM07], bisher fehlt jedoch eine ganzheitliche Betrachtung.

Die Untersuchung der existierenden Lösungen hat außerdem gezeigt, dass eine umfassende Lösung aufgrund des offenen Charakters mobiler Umgebungen nicht nur schwierig, sondern mit präventiven Maßnahmen alleine nicht realisierbar ist.

## 2.5 Reaktive mobile Sicherheitsarchitekturen

Reaktive Sicherheitsarchitekturen erweitern die Prävention vor Schutzzielverletzungen um die Kontrolle der Einhaltung der hierzu verwendeten Sicherheitsmaßnahmen. Dazu müssen reaktive Systeme im wesentlichen zwei Aufgaben erfüllen: die Erkennung von

Systemanomalien und die Reaktion auf solche. Sie adressieren damit die funktionalen Anforderungen */R2/ Überwachung* und */R3/ Reaktion*.

Die technische Basis für die Überwachung von Systemen und Erkennung möglicher Anomalien stellen sogenannte Intrusion Detection Systeme dar. Im folgenden wird daher zunächst eine kurze Einführung in deren Grundlagen gegeben, bevor konkrete reaktive Sicherheitsarchitekturen für mobile Umgebungen vorgestellt werden. Dabei wird wiederum zwischen Ansätzen zum Schutz mobiler Ad-hoc-Netze und solchen für infrastrukturelle Netze unterschieden. Erstgenannte stammen praktisch ausschließlich aus dem akademischen Bereich. Ihre Validierung stützt sich in aller Regel auf Simulationen. Heute verfügbare kommerzielle Entwicklungen hingegen konzentrieren sich auf die weiter verbreiteten infrastrukturellen Netze und setzen deutlich praxisorientiertere Lösungen um.

### 2.5.1 Grundlagen Intrusion Detection

Die Überwachung von Systemen und die Erkennung möglicher Bedrohungen ist Aufgabe sogenannter *Intrusion Detection Systeme (IDS)* [Bis03]. Unter einer *Bedrohung* (engl. *threat*) wird die potentielle Verletzung einer Sicherheitsrichtlinie verstanden. Damit diese Verletzung auftreten kann, müssen die Sicherheitsmechanismen des betrachteten Systems eine oder mehrere *Schwachstellen* (engl. *vulnerability*) aufweisen. Der Versuch, eine solche Schwachstelle tatsächlich auszunutzen, wird *Angriff* (engl. *attack*) oder auch *Einbruch* (engl. *intrusion*) genannt, wobei Angriffe nicht zwangsläufig erfolgreich sein müssen. Ein *Angreifer* (engl. *attacker* oder *intruder*) bedient sich hierzu immer häufiger automatisierter Programme, sogenannter *Angriffswerkzeuge* (engl. *attack tools* oder *exploits*). Akzeptiert man, dass ein Angriff nicht mit letzter Gewissheit ausgeschlossen werden kann, erscheint seine Erkennung mit Hilfe eines IDS sinnvoll, um die aus einem Angriff resultierenden Auswirkungen zu mildern. Hierzu durchsucht ein IDS im laufenden Betrieb eines Systems erhobene Daten, sogenannte *Audit-Daten*, nach Anzeichen für einen Angriff. Je nach Quelle der Audit-Daten unterscheidet man zwischen *Host Intrusion Detection Systems (HIDS)* und *Network Intrusion Detection Systems (NIDS)*. HIDS stützen sich ausschließlich auf lokal erhobene Informationen des Betriebssystems oder der Anwendungen eines Systems. Im Gegensatz hierzu analysieren NIDS den Netzverkehr.

Im folgenden wird zuerst ein kurzer historischer Überblick über die Wurzeln und Entstehung solcher Systeme gegeben, um dann genauer auf ihre Realisierung einzugehen. Für weitergehende Details sei auf [Amo99a] und [Bac00a] verwiesen.

#### Historische Ursprünge

Die Wurzeln der Intrusion Detection reichen zurück bis zu den traditionellen Audit-Systemen. *Audit-Systeme* generieren chronologische Aufzeichnungen von Systemer-

eignissen, um diese anschließend manuell zum Zwecke der Fehleranalyse, Systemoptimierung oder aber zur Abrechnung von Systemaktivitäten analysieren zu können. Im Rahmen der Sicherheitsinitiative des amerikanischen Verteidigungsministeriums in den 1970ern wurden erstmals formell Ziele definiert, solche Audit-Systeme zur Verbesserung der IT-Sicherheit einzusetzen, indem beispielsweise versuchte Zugriffsverletzungen oder unzulässige Systemaktivitäten aufgedeckt werden. Hierzu formulierte James P. Anderson als erster die Notwendigkeit, solche Audits maschinell durchzuführen und dabei redundante oder irrelevante Audit-Einträge (semi-)automatisch zu entfernen *(Audit Reduction)*. Sein Bericht [And80] aus dem Jahr 1980 gilt allgemein als die Keimzelle von Intrusion Detection. Die eigentliche Geburtsstunde des Intrusion Detection, wie wir es heute kennen, wird jedoch durch Dorothy Dennings Arbeit aus dem Jahre 1987 markiert [Den87]. Gemeinsam mit Peter Neumann hatte sie von 1984 bis 1986 ein theoretisches Modell für ein Echtzeit Intrusion Detection System entwickelt. Grundlage dieses Modelles war die Annahme, dass die Nutzung eines Computersystems charakteristische Muster aufweist, die sich statistisch erfassen lassen. *Anomalien*, d.h. Abweichungen von dieser „normalen“ Nutzung, werden einem Missbrauch gleichgesetzt. Sie begründete damit den *Anomalien erkennenden* Intrusion Detection Ansatz. Das Modell wurde 1986 bis 1992 in dem *Intrusion Detection Expert System (IDES)* [GNV+92] Prototyp umgesetzt und diente als Basis für eine Vielzahl weiterer Arbeiten und Systeme. Da Anomalien erkennende Verfahren selbständig das Nutzungsverhalten lernen und den aktuellen Gegebenheiten anpassen, mit anderen Worten sich selbständig konfigurieren, können sie ohne größeres Vorwissen eingesetzt werden. Zudem sind sie in der Lage, auch neue, zuvor unbekannte Angriffe aufzudecken. In der Praxis konnten sich die Systeme trotzdem nicht durchsetzen, was vor allem zwei Gründe hat. Zum einen ist die Generierung der Nutzungsprofile eine immens schwierige und rechenintensive Aufgabe, die sich zwangsläufig nur mit approximativen Verfahren lösen lässt. Es hat sich jedoch herausgestellt, dass die so erzeugten Nutzungsprofile zu ungenau sind und zu einer Vielzahl von False Positives führen. Zum anderen lassen sich Anomalien erkennende IDS angreifen, indem zunächst verschiedene Aktionen durchgeführt werden, die zwar noch dem normalen Nutzungsverhalten entsprechen, in der Folge jedoch das Nutzungsprofil des IDS derart ändern, dass der eigentliche Angriff nicht mehr entdeckt wird.

Der zweite Ansatz für Intrusion Detection, sogenannte *Missbrauch erkennende* Verfahren, verwendet Regeln, auch Signaturen genannt, die unzulässiges Verhalten exakt spezifizieren. Mit Hilfe dieser Regeln lassen sich die Audit-Daten sehr effizient und einfach nach verdächtigem Verhalten durchsuchen, was zu einer niedrigen False Positive Rate führt. Die Spezifikation der Signaturen erfordert jedoch ausgewiesene Spezialisten. Zudem müssen die Signaturen permanent erweitert werden, um neue oder modifizierte Angriffe erkennen und die Effektivität des IDS erhalten zu können. Um die Vor- und Nachteile der beiden Verfahren gegeneinander aufzuwiegen wurde schon früh die Kombination beider Ansätze gefordert. Eines der ersten Systeme, das einen solchen

hybriden Ansatz verfolgt, war MIDAS [SSHW88]. In der Praxis haben sich jedoch die reinen *Missbrauch erkennenden* Verfahren mehr durchgesetzt.

Einen weiteren Meilenstein in der Entwicklung des Intrusion Detection markiert der 1991 vorgestellte *Network System Monitor (NSM)* [HDL+90]. NSM war nicht nur der erste Versuch, Intrusion Detection auf heterogene Netzumgebungen auszuweiten, sondern es verwendete auch als erstes System Netzwerkpakete anstelle von Betriebssystemereignissen als Datenbasis. Seitdem wird zwischen *host-basiertem* und *netzwerkbasiertem* Intrusion Detection unterschieden. Ein Jahr später wurde mit dem *Distributed Intrusion Detection System (DIDS)* [SST92] das erste verteilte IDS vorgestellt, das hostbasiertes und netzwerkbasiertes Intrusion Detection vereint. DIDS verfolgt im Kern einen zentralisierten Ansatz: obwohl die Audit-Daten von mehreren im Netzwerk verteilten Sensoren erfasst werden, erfolgt die Analyse durch *ein* zentrales System. Dieser zentralisierte Ansatz wird von einer Vielzahl neuerer Entwicklungen verfeinert. Beispielsweise verwenden sowohl EMERALD [PN97] als auch AAFID [SZ00] hierarchisch organisierte Monitore, die die Ergebnisse von verteilten Sensoren resp. autonom agierender Agenten korrellieren. In der Praxis stellt dieser zentralisierte Ansatz jedoch einen Schwachpunkt hinsichtlich Skalierbarkeit und Robustheit dar. Sinnvoller erscheint die Kooperation unterschiedlicher autonomer IDS ohne zentralen Regisseur, wie sie 1996 erstmalig im *Cooperating Security Managers (CSM)* realisiert wird [WFP96]. In die gleiche Richtung weist auch das *Intrusion Detection Message Exchange Format (IDMEF)*, das die Definition eines Nachrichtenformats zum Informationsaustausch zwischen unterschiedlichsten IDS zum Ziel hat und auch die Einbindung bestehender Netzmanagementlösungen ermöglichen soll [DCF06].

Heute sind IDS ein fester Baustein beim Schutz lokaler Netze [ACF+00]. Kommerzielle Anbieter, wie etwa die bereits 1994 gegründete und unlängst von IBM übernommene Firma Internet Security Systems[2], bieten ein breites Portfolio an Intrusion Detection and Prevention Lösungen. Auch die Open Source Gemeinschaft stellt mit Projekten wie Snort[3] oder Prelude[4] praxistaugliche Systeme zur Verfügung. Trotzdem bleibt eine Vielzahl offener Probleme, an denen aktiv geforscht wird. Beispielhaft zu nennen ist hier etwa die Analyse von Angriffen, um präzisere Angriffssignaturen erstellen und so die Fehlerrate von IDS deutlich senken zu können. Auch macht die steigende Verbreitung von Intrusion Detection Systemen diese selbst zu einem lohnenden Angriffsziel, weshalb verstärkt über den Schutz des IDS selbst nachgedacht werden muss. Schließlich stellen neue Übertragungsmedien viele klassische IDS-Ansätze vor neue Herausforderungen. Zu erwähnen ist hier beispielsweise die Echtzeitverarbeitung des imensen Datenaufkommens moderner Hochgeschwindigkeitsnetze oder aber auch die speziellen

[2] http://www.iss.net/ (Abruf: August 2008)
[3] http://www.snort.org/ (Abruf: August 2008)
[4] http://www.prelude-ids.org/ (Abruf: August 2008)

Anforderungen, die sich aus mobilen Umgebungen ergeben. Im letztgenannten Bereich ist auch die vorliegende Arbeit angesiedelt.

#### Strategien zur Erkennung von Angriffen

Intrusion Detection Systeme lassen sich nach den von ihnen verwendeten Konzepten zur Analyse der Auditdaten und Erkennung von Angriffen in zwei Klassen unterteilen [Axe00]:

**Anomalieerkennung** fasst Intrusion Detection Systeme zusammen, die anhand charakteristischer Eigenschaften eines Systems Abweichungen vom erwarteten Systemverhalten aufspüren. Ihnen liegt die Annahme zugrunde, dass das legitime Verhalten von Benutzern und Systemen vorhersagbar und effizient modellierbar ist. Anomalieerkennung kann weiter in selbstlernende und spezifikationsbasierte Verfahren unterteilt werden. Bei *spezifikationsbasierten Verfahren* wird durch den Systemadministrator dem IDS das normale Systemverhalten in Form expliziter Regeln quasi einprogrammiert. Dies kann auf Grundlage deskriptiver statistischer Methoden geschehen oder auch in Form von Zustandsautomaten, die zulässige Systemzustände und Zustandsübergänge wiedergeben. *Selbstlernende anomalieerkennende IDS* beobachten zunächst in einer Lernphase das zu schützende System und generieren so ein Modell normalen Verhaltens. Die hierbei eingesetzten Techniken reichen von stochastischen Verfahren, wie etwa Schwellwertmetriken oder Markov-Modellen, bis hin zu künstlichen neuronalen Netzen oder genetischen Algorithmen, die der KI entstammen. Das Erstellen von Spezifikationen ist in der Regel nur für kleine Systeme mit klar abgegrenzter Funktionalität möglich. Existierende anomalieerkennende Systeme verfolgen daher meist einen selbstlernenden Ansatz. Die Anomalieerkennung ist dabei auf eine kontinuierliche Anpassung des zugrundeliegenden Systemmodells angewiesen, was sich in der Praxis jedoch als sehr schwierig erwiesen hat. Auch birgt der permanente Abgleich des Systemmodells mit dem tatsächlichen Systemverhalten die Gefahr, dass ein Angreifer durch sorgfältig gewählte Schritte das IDS so rekonfiguriert, dass ein Angriff nicht mehr als solcher erkannt wird. Der größte Nachteil anomalieerkennender IDS ist daher eine vergleichsweise hohe Fehlerrate. Auf der anderen Seite ermöglicht der Ansatz prinzipiell auch das Erkennen neuer Angriffe.

**Missbrauchserkennung** bezeichnet die Erkennung von Angriffen anhand bekannter Angriffsmuster, die als sogenannte Signaturen modelliert werden. Sie stellen damit den direkten Gegenpol zu spezifikationsbasierten Systemen dar. Missbrauchserkennende IDS verfügen ähnlich wie Virenscanner über eine Bibliothek von Angriffssignaturen und melden das Auftreten eines solchen Musters. Im Gegensatz zu anomalieerkennenden Systemen können sie also nur bereits bekannte Angriffe aufspüren. Der Pfle-

ge der Angriffsmuster kommt daher eine große Bedeutung zu. Sie erfordert in aller Regel fundiertes Expertenwissen zur Untersuchung bereits bekannter Angriffe und Angriffswerkzeuge oder auch zur Vorhersage neuer Bedrohungen durch die Analyse der eingesetzten Sicherheitsmechanismen und -protokolle. Die Definition möglichst präziser Angriffsmuster ist nicht einfach. Dennoch zeichnen sich missbrauchserkennende Systeme in der Praxis durch hohe Erkennungsraten aus.

Eine grundsätzliche Übersicht möglicher Ansätze zur Erstellung von Anomalie- und Missbrauchsmodellen findet sich bei Verwoerd und Hunt in [VH02].

### Aufbau von Intrusion Detection Systemen

Bishop gliedert die Architektur eines IDS wie in Abbildung 2.7 dargestellt in drei grundlegende logische Komponenten [Bis03]. Ein *Agent* beobachtet das zu überwachende System und sammelt hierbei relevante Daten, die er zur weiteren Analyse an den *Director* sendet. Dieser führt die eigentliche Angriffserkennung auf Grundlage der soeben beschriebenen Strategien durch. Entdeckt er einen Angriff, benachrichtigt er den sogenannten *Notifier*. Dieser entscheidet, ob und welche Aktionen in der Folge eingeleitet werden. Bei der Erwiderung eines Angriffs kann dabei zwischen passiven und aktiven Maßnahmen unterschieden werden [Bac00a]. Die *passive Erwiderung* eines Angriffs geht nicht über dessen Protokollierung und eine Benachrichtigung des Nutzers hinaus. Unter *aktiver Reaktion* oder auch *Intrusion Response* wird dagegen die automatische oder mit dem Nutzer abgestimmte Einleitung von Gegenmaßnahmen mit dem Ziel verstanden, die Auswirkungen des Angriffs abzuschwächen oder sogar gänzlich rückgängig zu machen.

Praktisch alle bekannten IDS orientieren sich an dieser Rahmenarchitektur, wobei einzelne Komponenten auch mehrfach auftreten können. Darüber hinaus kann zwischen *monolithischen* (engl. *stand-alone*) und *verteilten* (engl. *distributed*) IDS unterschieden werden, je nachdem ob die genannten logischen Komponenten auf einem lokalen System implementiert oder auf mehrere Knoten eines Netzwerks verteilt wurden.

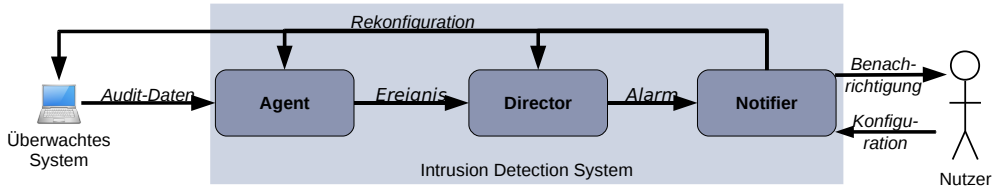



Abbildung 2.7: Allgemeine Architektur eines Intrusion Detection Systems

### 2.5.2 Intrusion Detection in mobilen Ad-hoc-Netzen

Das Gros der für mobile Ad-hoc-Netze vorgeschlagenen reaktiven Sicherheitsarchitekturen befasst sich mit der Überwachung des dynamischen Routings mittels eines Intrusion Detection Systems. Darüber hinaus legen einige Arbeiten auch schon Reaktionsmechanismen fest, wobei hier die Hauptschwierigkeit in der möglichst genauen Erkennung des aufgetretenen Zwischenfalls liegt. Eine weitere Gemeinsamkeit vieler Arbeiten ist ihr gerätezentrierter und kooperativer Ansatz. So wird die Funktionalität auf die einzelnen Endgeräte verteilt, die wie beim dynamischen Aufbau der Routing-Informationen auch gemeinsam Angriffe auf eben diese aufdecken sollen.

Eine der ersten Arbeiten zur Erkennung von Angriffen auf drahtlose Ad-hoc-Netze ist das von Zhang und Lee veröffentlichte Papier [ZL00], in der die prinzipielle Architektur für ein solches IDS vorgestellt wird. Sie bildet die Grundlage für eine Vielzahl weiterer Arbeiten, die einzelne Aspekte des ursprünglichen Modells weiterentwickeln. Im Zentrum der von Zhang und Lee vorgeschlagenen Lösung steht der einzelne Knoten eines mobilen Ad-hoc-Netzes, der selbst für seinen Schutz verantwortlich zeichnet, hierzu allerdings mit seinen Nachbarn kooperiert. Demonstriert wird der Ansatz zur Erkennung von Anomalien beim Aktualisieren von Ad-hoc-Routing-Tabellen. Eine experimentelle Evaluierung für verschiedene Ad-hoc-Routing-Protokolle unter Zuhilfenahme des Netzwerksimulators ns-2[5] findet sich in [ZLH03]. Die Abkehr von jeglichen zentralen Maßnahmen hin zu einer kooperativen Architektur autonomer Endgeräte stellt einen wichtigen Entwicklungsschritt dar. Aufgrund der – verglichen mit infrastrukturellen Netzen – gewandelten Struktur von Ad-hoc-Netzen erscheint dies nur folgerichtig. Die sich hieraus ergebenden Fragen, etwa nach der Verlässlichkeit oder Motivation von Kooperationspartnern, werden von den Autoren jedoch vernachlässigt. In [HL03] erweitern Huang und Lee das ursprüngliche System um eine regelbasierte Erkennung bekannter Angriffstypen, um nicht nur eine angemessenere Reaktion auf entdeckte Angriffe, sondern auch die möglichst genaue Identifikation des Angreifers zu ermöglichen. Zur Kompensation vorhandener Ressourcenbeschränkungen schlagen die Autoren ferner einen cluster-basierten Ansatz zur gemeinschaftlichen Überwachung eines Netzbereichs vor, bei dem die Kooperation mehrerer Knoten durch einen von ihnen dynamisch gewählten Cluster-Head koordiniert wird. Die damit einhergehende Sicherheitsproblematik wird erkannt, ohne jedoch Lösungsmöglichkeiten vorzuschlagen. Auch hier erfolgt die Validierung des Systems lediglich mittels Simulationen mit ns-2 und MobiEmu[6].

Die Idee, kooperierende Knoten in Clustern zu organisieren, wird in [DXL+06] erneut aufgegriffen, wiederum mit dem Ziel, die ressourcenintensive Erkennung von Angriffen an einen Cluster-Head zu delegieren und somit die einzelnen Knoten zu entlasten und die Performanz des Systems zu verbessern. Die Arbeit diskutiert detailliert verschiedene

[5] http://nsnam.isi.edu/nsnam/ (Abruf: August 2008)
[6] http://mobiemu.sourceforge.net/ (Abruf: August 2008)

Ansätze zur Bildung solcher Cluster auf Basis inhärenter Hierarchien der realen Systemumgebung, wie dies beispielsweise in militärischen Anwendungsszenarien der Fall ist, oder aber mittels spezieller Clustering-Protokolle. Eine Bewertung der Leistungsfähigkeit der vorgeschlagenen Lösung erfolgt wiederum auf Grundlage von Simulationen. Angesprochen werden auch essentielle Anforderungen an das System, wie die Bereitschaft der einzelnen Knoten zur Kooperation oder mögliche Angriffe durch einzelne arglistige Knoten. Lösungsvorschläge bleiben die Autoren jedoch wiederum schuldig.

Das gleiche Ziel, nämlich die Effizienz von Kooperationen durch eine strukturierte Organisation zu verbessern, verfolgt auch das in [SWP03] vorgestellte *ZBIDS (Zone-based Intrusion Detection System)*. Im Kern ein anomalieerkennendes IDS zur Überwachung des mobilen Ad-hoc-Routings, wird das vorhandene Netz zunächst anhand der physischen Positionen seiner Knoten in nichtüberlappende Zonen unterteilt. Im folgenden wird dann zwischen Intra- und Interzonen-Knoten unterschieden, je nachdem ob ein Knoten eine physische Verbindung zu Knoten anderer Zonen besitzt oder nicht. Wie bei den bereits vorgestellten Arbeiten versucht auch hier jeder Knoten eigenverantwortlich Angriffe zu erkennen, kann dabei aber mit anderen Knoten kooperieren. Intrazonen-Knoten können dazu jedoch lediglich per Broadcast Alarmmeldungen an die Knoten ihrer Zone versenden, während Interzonen-Knoten auch in der Lage sind, Alarmmeldungen über Zonengrenzen hinweg auszutauschen. Die Validierung erfolgt ebenfalls ausschließlich simulativ.

Das *Local Intrusion Detection System (LIDS)* ist eine weitere Konkretisierung von Lees und Zhangs ursprünglichem Ansatz [ACP+02]. Die Autoren diskutieren zudem erstmals das Problem der Verlässlichkeit kooperierender Knoten, indem sie den Terminus der „spontanen Gemeinschaften“ (engl. *spontaneous communities*) einführen. Hierunter verstehen sie eine Teilmenge der Knoten eines Ad-hoc-Netzes, die sich durch eine gemeinsame Vertrauensrelation definiert, und die bei der Erkennung von Angriffen zusammenarbeiten. Eine weitergehende Betrachtung dieser Vertrauensbeziehung fehlt jedoch, wie auch die Arbeit insgesamt eher oberflächlich bleibt. Über eine allgemeine Beschreibung der Systemarchitektur auf Grundlage mobiler Agenten und SNMP (Simple Network Management Protocol) hinaus wird nicht näher auf die technische Umsetzung eingegangen.

Erst in [PPM+03, PMPS04] erfolgt eine praktische Validierung des ursprünglichen Ansatzes. Der in Java implementierte Prototyp wird dabei im Laborumfeld hinsichtlich Performanz und Skalierbarkeit untersucht. Er unterstützt nicht nur die Erkennung von Angriffen auf das Ad-hoc-Routing-Protokoll, sondern demonstriert mit der Erkennung von Telnet Stepping-Stone-Angriffen auch den Einsatz auf Anwendungsebene. Überlegungen zur Verlässlichkeit der Kooperationsbeziehungen werden aber auch hier nicht angestellt.

Der in [YLY+04] vorgestellte flexible Sicherheitsentwurf (engl. *resilience-oriented security design*) konzentriert sich erstmals nicht nur auf die Netzwerkschicht. Stattdessen plädieren Yang et al. für eine vollständige und umfassende Lösung bestehend aus prä-

ventiven und reaktiven Sicherheitsmechanismen. Diese soll nicht nur den gesamten Protokollstapel umfassen, sondern auch mehrere Endgeräte zu einem kooperativen, mehrstufigen Sicherheitssystem integrieren. Für das unvermeidbare Auftreten von Fehlern und Angriffen soll so selbst im Falle der erfolgreichen Überwindung einer Maßnahme zumindest ein rudimentärer Schutz des Gesamtsystems sichergestellt werden. Als ersten Schritt zur Realisierung einer solchen Sicherheitsarchitektur analysieren die Autoren Angriffe und mögliche Gegenmaßnahmen auf die Netzwerk- und Sicherungsschicht. Darüber hinaus beschreiben sie verbleibende Herausforderungen für eine vollständige Umsetzung, wozu auch die Frage der Verlässlichkeit von kooperierenden Knoten zählt.

Die in [Kar03] vorgestellte *Sicherheitsarchitektur für Mobile Ad hoc Netzwerke (SAM)* stellt wohl den ersten Versuch dar, singuläre Schutzmechanismen für mobile Ad-hoc-Netze zu einer umfassenden und aufeinander abgestimmten Sicherheitslösung zusammenzufassen, die sämtliche relevanten Fragestellungen adressiert und ohne unrealistische initiale Annahmen auskommt. SAM ist aus drei Komponenten aufgebaut. Das hybride *Secure Dynamic Source Routing (SDSR)* verhindert eine große Bandbreite von Fehlverhalten bei der Wegewahl. Für die Erkennung möglicher Manipulationen arbeitet es eng mit dem kooperativen Intrusion Detection System *MobIDS* zusammen, das in Struktur und Funktionalität auf dem Basisentwurf von Zhang und Lee [ZL00, ZLH03] aufbaut. Beide Komponenten greifen auf das ebenfalls entwickelte Identifikationsverfahren MANET-IDs zurück, beispielsweise um nicht regelkonforme Knoten eindeutig zu identifizieren und von weiteren Interaktionen auszuschließen. *MANET-IDs* weisen jedem Knoten den öffentlichen Teils eines asymmetrischen Schlüsselpaars als eindeutige und unveränderliche Identität und damit als Wiederkennungsmerkmal zu. Um Änderungen dieses Identikators zu unterbinden, werden die Schlüssel durch eine vertrauenswürdige dritte Instanz zertifiziert. Der in [BEGA03] beschriebene Ansatz statistisch eindeutiger und kryptographisch verifizierbarer Kennzeichen (engl. *Statistically Unique and Cryptographically Verifiable (SUCV) Identifier*) wird schließlich zur Verknüpfung des Identifikators mit der konkreten Adresse eines Netzknotens genutzt. Um der sich hieraus ergebenden Gefährdung des Datenschutzes zu begegnen, wird die Verwendung von Pseudonymen mittels sogenannter *Pseudo-MANET-IDs* vorgeschlagen. Um einer Weitergabe gültiger MANET-IDs vorzubeugen, beschreibt Kargl außerdem einen sogenannten Identitätsrückruf, um solche Knoten zu sperren, die das MobIDS als Angreifer identifiziert hat.

Kargls Arbeit versucht sich als erste an einer ganzheitlichen Sicht der vielen Teilaspekte von Sicherheit in mobilen Umgebungen. Sie konzentriert sich dabei ausschließlich auf mobile Ad-hoc-Netze, für die ein detailliertes Sicherheitsrahmenwerk geschaffen wird. Daran lassen sich auch die beiden wesentlichen Kritikpunkte an der Arbeit festmachen. So werden infrastrukturelle Netze gänzlich außer Acht gelassen, obwohl sie in der Praxis derzeit den Standardfall darstellen. Zum anderen ist die ausschließlich simulative Validierung der vorgeschlagenen Lösung, wie Kargl selbst anmerkt, nur bedingt aussagekräftig, da sie wichtige praxisrelevante Aspekte wie etwa den Ressourcenbedarf

der Sicherheitsmechanismen außer Acht lässt. Dennoch stellt die Arbeit einen wichtigen Schritt hin zu einem ganzheitlichen Sicherheitskonzept für mobile Umgebungen dar, indem sie präventive und reaktive Sicherheitsmechanismen miteinander integriert. Außerdem spricht sie explizit das Problem der Verlässlichkeit von Kooperationspartnern an und macht hierfür konkrete Lösungsvorschläge. So wird etwa ein Schwellwertverfahren eingeführt, um die Diskreditierung einzelner Knoten durch koalierende Angreifer im Rahmen der globalen Bewertung zu erschweren.

Ein weiterer Versuch, die Angriffserkennung in MANETs auf eine breitere Basis zu stellen, findet sich in [SBC+05]. Die vorgeschlagene generalisierte Architektur eines kooperativen IDS ermöglicht ebenfalls die Erkennung von Angriffen auf unterschiedlichen Ebenen des ISO/OSI-Modells. Im Zentrum der vorgeschlagenen Architektur steht wiederum der einzelne Knoten, der selbst für seinen Schutz verantwortlich zeichnet. Um jedoch eine erweiterte Schutzlinie einzuziehen, werden die einzelnen Knoten ähnlich den bereits erwähnten cluster-basierten Ansätzen in einer dynamisch anpassbaren Hierarchie organisiert. In dieser sammeln die Blattknoten relevante Daten, aggregieren und analysieren sie und reichen die so reduzierten Daten inkrementell an darüber liegende Cluster-Heads weiter. Die Cluster-Heads werden dynamisch nach Kriterien wie etwa Position, Leistungsfähigkeit, Sicherheit oder Vertrauenswürdigkeit ausgewählt. Die Weitergabe von Sicherheitsrichtlinien erfolgt umgekehrt von den Cluster-Heads hinab zu den einfachen Knoten. Die Problemanalyse und Beschreibung der Architektur spricht zwar viele relevante Fragen an, bleibt insgesamt aber abstrakt und informell. So werden die notwendigen Komponenten und ihre Funktionalität nur grob umrissen. Auch die Validierung, die sich ausschließlich auf die Betrachtung eines missbrauchserkennenden IDS für drei exemplarische Anwendungsszenarien stützt, fällt eher vage aus. Konkrete Vorschläge für eine praktische Realisierung fehlen.

### 2.5.3 Intrusion Detection in mobilen infrastrukturellen Netzen

Die Mehrzahl der auf mobile infrastrukturelle Netze spezialisierten reaktiven Sicherheitslösungen ist kommerziellen Ursprungs. Tabelle 2.2 auf der nächsten Seite fasst die bekanntesten Anbieter und Produkte zusammen. Die Arbeiten konzentrieren sich dabei auf die Überwachung und Sicherung der Netzzugangsschicht.

#### Kommerzielle Systeme

Heutige kommerzielle Lösungen zum Schutz mobiler Netze konzentrieren sich praktisch ausschließlich auf Wireless LANs nach dem IEEE 802.11 Standard. Nur langsam erweitern die Anbieter ihr Portfolio um spezielle Produkte für Bluetooth. Bisher überwiegt jedoch die Einschätzung von Wireless Personal Area und Ad-hoc-Netzen als potentiell gefährlich, weshalb ihre Aktivierung und Nutzung durch die meisten Lösungen reglementiert wird. Überhaupt ähneln sich die einzelnen Produkte sehr in Konzeption und

| Anbieter | Produkt | Erläuterung |
|---|---|---|
| AirDefense | AirDefense Enterprise | Wireless IDS |
| | AirDefense Personal | Policy Enforcement Tool |
| | AirDefense BlueWatch | Bluetooth Monitor Software |
| AirMagnet | AirMagnet Enterprise | Wireless IDS |
| | AirMagnet StreetWISE | Policy Enforcement Tool |
| | AirMagnet BlueSweep | Bluetooth Monitor Software |
| Aruba | ArubaOS Wireless Intrusion Protection | Wireless Management- und IDS-Lösung |
| Cisco | Secure Wireless Solution | Wireless Management- und IDS-Lösung |
| Madge | WLAN Probe Monitor | WIDS Server |
| | WLAN Probe 2 | WIDS Sensor |
| Network Instruments | Observer Wireless | WLAN Monitor Software |
| Newbury Networks | WiFi Watchdog | Wireless IDS |
| WildPackets | OmniPeek | WLAN Monitor Software |

Tabelle 2.2: Auswahl kommerzieller Wireless Sicherheitslösungen

Funktionsweise. Abbildung 2.8 auf der nächsten Seite zeigt die typische dreigeteilte Systemarchitektur kommerzieller *Wireless Intrusion Detection Systeme (WIDS)*, die sich an der in Kapitel 2.5.1 auf Seite 44 beschriebenen Rahmenarchitektur orientiert.

Sie umfasst neben Sensoren einen zentralen WIDS Server sowie eine WIDS Management Konsole zur Administration des Systems. Die Sensoren auf Basis herkömmlicher Access Points oder aber auch spezieller Hardware ermöglichen die Überwachung der Funkschnittstelle, indem die übertragenen Daten mitgeschnitten, einer ersten Verarbeitung unterzogen und dann an den WIDS Server übermittelt werden. Hier erfolgt die eigentliche Analyse und Angriffserkennung auf Basis einer Datenbank individuell anpassbarer Sicherheitsrichtlinien. Die Kommunikation zwischen den einzelnen Komponenten erfolgt auf IP-Basis, in der Regel in einem eigenen kabelbasierten Managementnetz. Nicht zuletzt aus Kostengründen ist zudem der Trend zu beobachten, die Systeme in bereits vorhandene Netzmanagementlösungen zu integrieren.

Ergänzend bieten einige Hersteller außerdem spezielle *Policy Enforcement Tools* an, die ähnlich einem Virenscanner auf mobilen Endgeräten installiert die Konformität ihrer Konfiguration mit zentral kontrollierten Richtlinien sicherstellen sollen. Der Charakter dieser Werkzeuge ist dabei eher präventiv und sie sind bisher nicht in die WIDS-Architektur integriert.

Eine Bewertung der angebotenen Lösungen gestaltet sich schwierig. So unterstützen alle Anbieter die Erkennung der gängigen Angriffe auf infrastrukturelle WLANs, wie et-

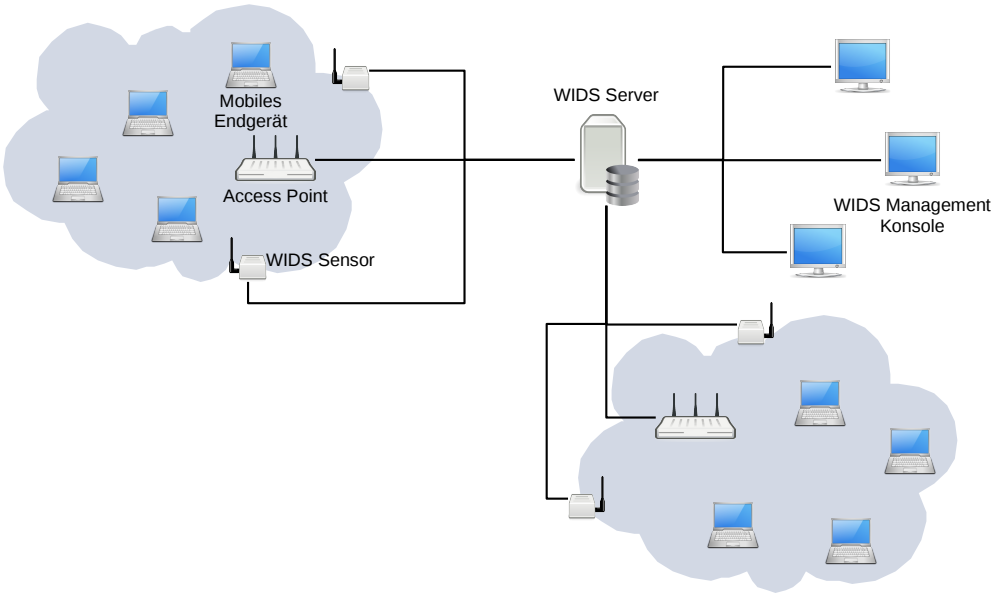



Abbildung 2.8: Systemarchitektur kommerzieller Wireless IDS

wa Rogue Access Points, Denial-of-Service (DoS) oder Man-in-the-Middle (MitM). Auch die Lokalisierung eines Angreifers bieten die meisten Systeme. Ein direkter Vergleich der jeweiligen Erkennungsleistung ist dennoch kaum möglich, da diese von einer Vielzahl von Faktoren abhängt. Hierzu gehören neben der allgemeinen Beschaffenheit der Einsatzumgebung Parameter, wie die Anzahl und Platzierung der Sensoren, die Anzahl der überwachten Übertragungskanäle oder auch das zu überwachende Frequenzband. Darüber hinaus hängt die Qualität der Angriffserkennung wesentlich von der Umsetzung des WIDS Servers ab. Diese wird neben der Art und Weise, wie die einzelnen Komponenten miteinander kombiniert werden, vor allem durch die Algorithmen und Regeln zur Identifikation eines Angreifers bestimmt. Eine genaue Analyse und Bewertung ist hierbei leider unmöglich, da die Details der jeweiligen Implementierungen nicht offen gelegt werden. Einschlägige Tests bewerten stattdessen die Unterstützung des Nutzers bei Aspekten wie etwa der Verwaltung von Alarmmeldungen oder der Konfiguration von Sicherheitsrichtlinien [Bul06b]. Die Forderung an die Hersteller, ihre Angriffssignaturen transparent zu machen und sich auf eine ganzheitliche Produktentwicklung zu konzentrieren [Bul06a], erscheint daher folgerichtig. Das Hauptaugenmerk sollte hierbei auf der Verwaltung von Signaturen, Angriffen und möglichen Gegenmaßnahmen liegen.

**Akademische Lösungen**

Wissenschaftliche Publikationen zum Schutz mobiler Infrastrukturnetze widmen sich meist einzelnen Aspekten des Problems. So ermöglichte der Distributed Wireless Security Auditor (DWSA) [BNPDS04] bereits frühzeitig die Lokalisierung von Rogue Access Points. Der Prototyp implementiert hierzu nicht nur Verfahren zur Erkennung solcher Angriffe, sondern auch zur Lokalisierung des Angreifers mittels Trilateration. Die vorgeschlagene Systemarchitektur entspricht im wesentlichen der heutiger kommerzieller Systeme, verzichtet aber auf dedizierte Sensoren. Stattdessen wird auf allen Endgeräten ein Sensor-Client installiert, der Netzinformationen wie MAC-Adressen, GPS-Koordinaten, IEEE 802.11 Protokolldetails oder aber auch Statistiken zu den Signalstärken der ihm benachbarten Geräte sammelt. Diese Informationen werden an einen zentralen Auditserver gesendet, der die Daten korreliert, gemäß vorgegebener Sicherheitsrichtlinien analysiert und grafisch aufbereitet. Darüber hinaus implementiert der Auditserver grundlegende Mechanismen zur Sicherung der Kommunikation mit dem Client. Die Frage der Verlässlichkeit eines Clients wird jedoch ebenso außer Acht gelassen wie die der Verfügbarkeit.

[SLO04] adaptiert das in [ZL00] erstmals vorgestellte MANET-IDS für infrastrukturelle mobile Netze. Wie im ursprünglichen System überwacht ein IDS-Agent auf jedem Endgerät die lokalen Aktivitäten. Darüber hinaus tauschen sich die lokalen Agenten untereinander aus, um gemeinsam Angriffe zu erkennen und koordinierte Reaktionen zu ermöglichen. Der ursprünglich anomaliebasierte Ansatz wurde außerdem um eine signaturbasierte Angriffserkennung ergänzt. Die Evaluierung erfolgt mittels einer Fallstudie für einen Man-in-the-Middle-Angriff anhand einer Linux-basierten Prototypimplementierung.

In [ACL06] wird eine verteilte Architektur modelliert, um Jamming-Angriffe von herkömmlichen Netzüberlastungen zu unterscheiden. Die Autoren schlagen hierzu ebenfalls die Einrichtung eines Sensor-Clients auf jedem Endgerät vor, der für die Netzverbindung relevante Daten erfasst und in Ereignislisten sammelt. Durch Korrelation dieser Ereignislisten erhält man einen detaillierten Überblick des Zustands des gesamten Netzes. Im Gegensatz zum DWSA erfolgt die Auswertung dabei nicht zentral, sondern durch jeden einzelnen Client. Der hierfür notwendige Austausch der Ereignislisten erfolgt mittels Broadcast. Darüber hinaus beschreiben die Autoren die Nutzung ihres Systems als Anreiz für die Einhaltung einer zugesicherten Dienstqualität. Die Widerstandsfähigkeit des vorgeschlagenen Systems gegen Angreifer, die ihren persönlichen Nutzen maximieren wollen, untersuchen die Autoren spieltheoretisch. Ihre Analyse berücksichtigt dabei bewusst keine Sabotage, die das Netz vollständig zum Erliegen bringen würde. Dies und der ineffiziente Austausch der Ereignislisten mittels Broadcast lässt das vorgeschlagene System nur bedingt praxistauglich erscheinen.

Wenn auch nicht primär zur Erkennung von Angriffen in WLANs gedacht, soll schließlich noch das jüngst von Microsoft Research entwickelte System WiFiProfiler

[CPZ06] vorgestellt werden. Es handelt sich dabei um ein kooperatives System zur Diagnose von Netzwerkproblemen in Wireless LANs, das ebenfalls durch den vollständigen Verzicht auf jegliche Infrastruktur hervorsticht. Im Gegensatz zu dem in [ACL06] beschriebenen Ansatz ist es jedoch deutlich praxisorientierter und als prototypische Implementierung für Windows XP verfügbar. Die Grundidee läßt sich wie folgt zusammenfassen. Schlägt der Verbindungsversuch eines WLAN-Clients mit einem Access Point fehl, bittet er über Ad-hoc-Verbindungen benachbarte Clients, ihm bei der Problembeseitigung behilflich zu sein. Auf den Endgeräten wird dazu eine Software bestehend aus Sensor-, Kommunikations- und Diagnose-Komponente benötigt. Die Sensor-Komponente erfasst relevante Daten über den Zustand und die Konfiguration der Netzverbindung eines Endgeräts. Die Kommunikations-Komponente realisiert die Ad-hoc-Kommunikation mit anderen Geräten, wohingegen die Diagnose-Komponente die Informationen der helfenden Clients analysiert und hieraus eine Problemlösung ableitet. Der Ansatz von WiFiProfiler hebt sich in mehreren Punkten von den bisher diskutierten ab. So verzichtet er vollständig auf jegliche Infrastruktur und setzt stattdessen ausschließlich auf die Kooperation mit anderen Endgeräten. Damit einher geht auch ein Paradigmenwechsel von der zentralen Kontrolle durch den Netzwerkadministrator hin zu einer mehr nutzer- bzw. endgerätezentrierten Sichtweise. Offen bleiben jedoch die daraus resultierenden Anforderungen insbesondere in punkto Sicherheit und Kooperationsbereitschaft, die von den Autoren lediglich knapp umrissen werden.

## 2.6 Angriffstolerante mobile Sicherheitsarchitekturen

Rasmusson und Jansson untersuchen in [RJ96] erstmalig, warum herkömmliche Sicherheitsmechanismen wie Passwörter, Zugangskontrolle etc. alleine nicht zum Schutz offener Systeme ausreichen. So lässt sich byzantinisches Fehlverhalten [LSP82] einzelner Systemkomponenten niemals gänzlich ausschließen. Diese können sich vielmehr willentlich oder unwillentlich fehlerhaft und darüber hinaus auch völlig inkonsistent verhalten. Der in [YLY+04] beschriebene flexible Sicherheitsentwurf folgt dieser Argumentation, indem er einen Paradigmenwechsel von konventioneller Angriffsvermeidung und -erkennung hin zur Angriffstoleranz auf Grundlage weicher Sicherheitsmechanismen fordert.

In [RJ96] werden soziale Kontrollmechanismen zum Schutz offener Systeme unter dem Begriff *Soft Security* zusammengefasst, um sie von traditionellen Sicherheitsmechanismen (*Hard Security*) abzugrenzen.

Bezogen auf mobile Ad-hoc-Netze erwartet Soft Security nicht nur die Existenz ungewollter Eindringlinge in ein Netz, sondern akzeptiert die Tatsache, dass sich dies in der Praxis nicht verhindern lässt. Stattdessen versuchen Soft-Security-Mechanismen böswillige oder fehlerhafte Komponenten zu identifizieren und reguläre Komponenten vor weiterem Schaden zu bewahren, indem nicht konforme Knoten zum „Umden-

ken“ bewegt oder aus dem Netz ausgeschlossen werden. Sie ermöglichen damit die Durchsetzung semantischer Sicherheit gemäß Anforderung /R7/ in Kapitel 2.3.3 auf Seite 28.

Bei den hierzu in der Literatur vorgeschlagenen Lösungen, die alle im Bereich der mobilen Ad-hoc-Netze angesiedelt sind, lassen sich die beiden im Folgenden beschriebenen Ansätze unterscheiden.

### 2.6.1 Anreizorientierte Verfahren

Anreizorientierte Verfahren versuchen egoistisches Verhalten in mobilen Ad-hoc-Netzen unrentabel zu machen. Ein Beispiel hierfür ist das von Buttyán und Hubaux entwickelte Nuglets System [BH00, BH03].

*Nuglets* sind eine virtuelle Währung zur gegenseitigen Verrechnung von Diensten in mobilen Ad-hoc-Netzen, wie etwa der Weiterleitung von Daten. Die Autoren diskutieren hierfür zwei Bezahlmodelle: für den Versand von Datenpaketen (*Packet Purse Model*) oder für deren Empfang (*Packet Trade Model*). Außerdem beschreiben sie als rationale Grundlage für oder wider die Diensterbringung ein Verfahren, die damit verbundenen Kosten dem persönlichen Nutzen eines Knotens gegenüberzustellen.

Die Arbeit stellt eine interessante Variante zur Vermeidung egoistischen Verhaltens dar, weist im Detail jedoch einige Probleme auf. So setzen die Autoren bei den Endgeräten manipulationssichere Hardware voraus, um unzulässige Eingriffe durch den Nutzer zu unterbinden, was in der Praxis nur schwer zu realisieren sein dürfte.

Auch weisen beide Bezahlmodelle Schwächen auf. So können einem Knoten im *Packet Trade Model* durch Zusenden vieler Datenpakete sämtliche *Nuglets* und somit jegliche weitere Kommunikationsmöglichkeit entzogen werden. Im *Packet Purse Modell* hingegen ist es für Knoten in exponierter Lage schwierig, genügend Nuglets für den von ihnen generierten Datenverkehr zu verdienen.

### 2.6.2 Reputationsbasierte Verfahren

Reputationsbasierte Lösungen fügen der anzreizorientierten Herangehensweise eine weitere Dimension hinzu, indem sie Sanktionen gegen sich nicht konform verhaltende Knoten einführen. Hierzu bewerten sich Knoten eines mobilen Ad-hoc-Netzes gegenseitig, um eine Entscheidungsgrundlage für zukünftige Interaktionen zu schaffen.

Nähere Details zu den Grundlagen von Reputationssystemen finden sich in Kapitel 6. An dieser Stelle sollen lediglich exisitierende reputationsbasierte Systeme zum Schutz des Ad-hoc-Routings exemplarisch beschrieben werden.

Wohl als erste kombinieren Marti et al. in ihrer Arbeit [MGLB00] ein IDS namens *Watchdog* mit einem rudimentären Reputationssystem genannt *Pathrater*, um egoistisches Verhalten in einem Ad-hoc-Netz zu erkennen und seine Auswirkungen einzudämmen. Das beschriebene Vorgehen lässt sich grundsätzlich an beliebige Routing-Protokolle

adaptieren, wobei die Originalarbeit die Erweiterung des DSR-Protokolls beschreibt. Der Watchdog überwacht dabei seine Nachbarknoten, ob diese innerhalb einer gewissen Zeitspanne wie vereinbart Datenpakete weiterleiten. Ist dies nicht der Fall, benachrichtigt er den Absender des betreffenden Pakets, wobei eine gewisse Fehlertoleranz berücksichtigt wird. Die Pathrater-Komponente kombiniert die so erhaltenen Meldungen über das Fehlverhalten einzelner Knoten mit Informationen über die Verlässlichkeit von Links, um so möglichst optimale Routen zu wählen. Die Autoren haben im Rahmen von Simulationen gezeigt, dass ihr Ansatz in der Tat signifikant vor einfachem egoistischen Verhalten schützt. Im Detail konstatiert aber bereits die Originalpublikation einige gravierende Schwächen, etwa im Hinblick auf adaptive und kooperierende Angreifer oder auch die Möglichkeit Knoten zu verleumden. Ein weiterer grundsätzlicher Kritikpunkt ist das Fehlen nennenswerter Repressalien [Kar03]. Zwar informiert der Watchdog den Absender einer Nachricht über das Fehlverhalten eines bestimmten Knotens. In der Folge führt dies aber lediglich dazu, dass dieser Knoten nicht mehr von diesem ihm offensichtlich missliebigen Knoten „behelligt“ wird. Ansonsten hat der fehlerhafte Knoten keine weiteren Nachteile zu befürchten.

Buchegger und Boudec nehmen dies zum Anlass, die Grundidee des Watchdog/Pathrater-Ansatzes in ihrem System *CONFIDANT (Cooperation of Nodes: Fairness in Dynamic Ad hoc NeTworks)* fortzuführen, indem sie unzuverlässige Knoten erkennen und konsequent isolieren [BB02]. Ihr Ansatz nutzt dazu nicht nur das selbst beobachtete Verhalten, sondern lernt auch aus den Beobachtungen anderer Knoten. Im einzelnen besteht CONFIDANT aus vier Komponenten. Analog zum bereits vorgestellten Watchdog-Ansatz überwacht der sogenannte *Monitor* seine direkte Nachbarschaft. Bei Unregelmäßigkeiten verständigt er das *Reputation System*. Dieses ordnet jedem Knoten eines MANETs eine individuelle Bewertung basierend auf eigenen Beobachtungen und den Informationen anderer Knoten zu. Diese werden durch den sogenannten *Trust Manager* gewichtet, wobei eigenen Beobachtungen eine höhere Relevanz zugesprochen wird als denen von Bekannten oder gar fremden Knoten. Unterschreitet die so ermittelte Bewertung eines Knotens einen bestimmten Toleranzwert, initiiert der *Path Manager* die Suche nach einem zuverlässigeren Weg und sperrt die Weiterleitung von Datenpaketen des betroffenen Knotens. Auch für CONFIDANT lassen sich einige wesentlichen Schwachstellen feststellen. So bleiben auch beim hier verfolgten Monitor-Ansatz adaptive und kollaborierende Angreifer unberücksichtigt. Überhaupt werden mögliche Angriffe auf das CONFIDANT-System selbst, wie etwa das Ändern der Knotenidentität, nicht betrachtet. Ein weiteres Problem stellt das ausschließliche Sammeln negativer Bewertungen dar. Um die permanente Isolation zu Unrecht beschuldigter Knoten zu verhindern, muss von Zeit zu Zeit die Historie gelöscht werden, so dass Angreifer immer wieder Zugang zum Netz erhalten.

Einen Beitrag zur weiteren Systematisierung reputationsbasierter Sicherheitsarchitekturen liefert CORE [MM02]. Auch hier wird das Verhalten von Knoten in einem MANET bewertet, wobei negative Bewertungen Sanktionen nach sich ziehen und untereinander

kommuniziert werden. Der wesentliche Beitrag der Arbeit liegt in der genauen Analyse des Reputationskonzepts und der ihm zugrundeliegenden Algorithmen zur Bewertung der Verlässlichkeit eines Knotens. Auch definieren die Autoren ein komplettes Rahmenwerk für ein System zur Durchsetzung kooperativen Verhaltens in mobilen Ad-hoc-Netzen, das in seiner konkreten Ausgestaltung jedoch weitestgehend abstrakt bleibt. Damit ist auch keine simulative Evaluation des Ansatzes möglich, wie sie die anderen Arbeiten vornehmen.

Das in [AHNRR02] vorgestellte dreistufige Protokoll *On-demand Secure Routing Protocol Resilient to Byzantine Failures (OSRP)* kombiniert als erstes reputationsbasierte Mechanismen mit kryptographischen Primitiven. Es realisiert ein robustes bedarfsorientiertes Routing-Verfahren, das auch byzantinisches Verhalten von Knoten berücksichtigt. Dazu definiert es eine Verlässlichkeitsmetrik, die Verbindungen zwischen zwei beliebigen Knoten auf Grundlage bereits gemachter Erfahrungen bewertet. Hierbei gilt, je unzuverlässiger eine Verbindung ist, desto höher ist die mit ihr assoziierte Gewichtung. Jeder Knoten nutzt eine solche Liste gewichteter Verbindungen, um bei der Wegewahl unzuverlässige Pfade zu vermeiden. Die Wegewahl ist dazu in drei sukzessiv ablaufende Phasen unterteilt. Bei der *Wegefindung* (engl. *Route Discovery*) sendet ein Quellknoten eine Routing-Anfrage an alle seine Nachbarknoten, um einen Pfad zu einem Zielknoten zu bestimmen. Per Flooding wird der Route Request durch das Netz bis zum Zielknoten propagiert, dessen Antwort ebenfalls per Flooding zurück an den Quellknoten gesendet wird. Die Verteilung der Routing-Nachrichten per Broadcast ist zwar äußerst ineffizient, wird aber benötigt, um Angriffe durch Einzelne zu unterbinden. Als Schutz vor möglichen Denial-of-Service-Angriffen werden allerdings nur autorisierte Routing-Anfragen zugelassen, d. h. ein Knoten leitet nur solche Anfragen weiter, die tatsächlich von einem seiner Nachbarn stammen. Die hierfür notwendige Authentisierung erfolgt mittels digitaler Signaturen. Die so ermittelte Route zwischen Quell- und Zielknoten wird dann im Rahmen der *byzantinischen Fehlererkennung* (engl. *Byzantine Fault Detection*) auf fehlerhafte Verbindungen überprüft. Das hierbei verwendete adaptive Probing sendet dazu spezielle Testpakete an zufällig gewählte Knoten entlang einer Route, die von den jeweiligen Empfängern quittiert werden müssen. Überschreitet die Anzahl der nicht korrekt quittierten Testpakete einen bestimmten Schwellwert, wird für die betroffene Route ein Fehler konstatiert. Die genaue Position der fehlerhaften Verbindung auf dieser Route wird dann mittels einer binären Suche ermittelt. Unzuverlässige Links können damit nach $O(\log n)$ aufgetretenen Fehlern identifiziert werden, wobei $n$ die Länge der jeweiligen Route ist. Zum Schutz dieses Mechanismus werden wiederum kryptographische Primitive eingesetzt, wobei aus Effizienzgründen symmetrische Verschlüsselungsverfahren verwendet werden. Die beteiligten Knoten müssen daher vorab paarweise geheime Schlüssel vereinbart haben. Der so identifizierte fehlerhafte Link wird dann im finalen Schritt der *Verbindungsbewertung* (engl. *Link Weight Management*) neu eingestuft. Diese Phase verwaltet eine Liste der bei der Fehlererkennung auffällig gewordenen Verbindungen und „bestraft“ diese, indem die der Verbindung zugeordnete

Gewichtung verdoppelt wird. Ferner verknüpft die Verbindungsbewertung jede Verbindung mit einem Zähler, bei dessen Ablauf die gespeicherte Gewichtung halbiert und somit der Verbindung die Möglichkeit zur Wiedereingliederung in den Routing-Prozess gegeben wird. Die Liste mit den Bewertungen einzelner Verbindungen wird, wie bereits beschrieben, bei der Wegefindung zur Vermeidung unzuverlässiger Pfade verwendet. OSRP hebt sich von den übrigen in diesem Abschnitt vorgestellten Verfahren ab, da es konkrete Maßnahmen gegen sich selbst gerichtete Angriffe vorschlägt. Die hierfür vorausgesetzte Existenz einer permanent verfügbaren Public-Key-Infrastruktur sowie paarweise vereinbarter symmetrischer Schlüssel zwischen Quell- und Probe-Knoten erscheint in der Praxis jedoch fragwürdig. Ein grundsätzlicher Kritikpunkt ist auch die Erkennung defekter Links anstatt der Identifikation der hierfür verantwortlichen Knoten. Es findet daher auch keine Bestrafung dieser Knoten statt, sondern nur eine Vermeidung unzuverlässiger Routen, womit egoistischen Knoten in die Hände gespielt wird.

## 2.7 Zusammenfassung und Diskussion

Tabelle 2.3 auf Seite 59 fasst die Ergebnisse der Untersuchung existierender Sicherheitsarchitekturen für mobile Umgebungen zusammen und stellt die einzelnen Systeme den in Kapitel 2.3 ermittelten Anforderungen gegenüber.

Während für die funktionalen Anforderungen lediglich angegeben wird, ob sie von der jeweiligen Arbeit adressiert werden oder nicht, wird für die nichtfunktionalen Anforderungen der Versuch einer rudimentären qualitativen Aussage unternommen. Dabei wird unterschieden, ob die Erfüllung einer Anforderung im Fokus einer Arbeit stand (Bewertung „gut“), ob die Autoren die entsprechende Fragestellung zumindest erkannt haben (Bewertung „durchschnittlich“) oder ob sie die Problematik gänzlich ignorieren (Bewertung „unzureichend“). Als praxistauglich wird eine Arbeit eingestuft, wenn sie zumindest prototypisch implementiert wurde, was für die Mehrzahl der reaktiven Architekturen für mobile Ad-hoc-Netze nicht der Fall ist. WEP wurde aufgrund seiner Sicherheitsmängel ebenfalls als nicht mehr praxistauglich eingestuft.

Es zeigt sich, dass alle Arbeiten nur einzelne Aspekte des Gesamtproblems betrachten. So konzentrieren sich die Arbeiten im Bereich MANETs im wesentlichen auf Schicht 3 des OSI-Referenzmodells. Dabei verfolgen sowohl die präventiven als auch die reaktiven Maßnahmen zum Schutz der Wegefindung und der Paketvermittlung prinzipiell einen dezentralen verteilt-kooperativen Ansatz, wobei die Kooperationsgemeinschaften zur Verbesserung der Performanz oftmals hierarchisch in Clustern oder Zonen organisiert werden [AW07]. Aufgrund der Beschränkung auf die Vermittlungsschicht und der fehlenden Berücksichtigung bestehender mobiler infrastruktureller Netze, erfüllen die meisten Systeme jedoch nicht die Anforderung der Universalität. Lediglich [SWP03, SBC+05, YLY+04] fordern einen übergreifenden Ansatz, bleiben bei

der Beschreibung einer solchen Systemarchitektur jedoch sehr vage, weshalb sie eine durchschnittliche Wertung erhalten.

Dies gilt auch für SAM, obwohl das in [Kar03] vorgeschlagene System ausschließlich auf mobile Ad-hoc-Netze abzielt. Gegenüber den übrigen betrachteten Systemen nimmt es dennoch eine herausgehobene Stellung ein. So stellt es wohl den ersten Versuch einer ganzheitlichen Betrachtung der Sicherheitsproblematik verbunden mit einem konkreten Systementwurf dar. Dieser wird jedoch, wie in diesem Bereich allgemein üblich, ausschließlich simulativ validiert, weshalb keine belastbaren Aussagen zur Praxistauglichkeit des Vorschlags gemacht werden können. Kargls Arbeit verdeutlicht dabei, dass präventive Maßnahmen sicherlich die Basis einer soliden Sicherheitsarchitektur darstellen. Sie zeigt aber auch ganz klar ihre Grenzen auf, etwa bei der Durchsetzung semantischer Sicherheit. Kargl verknüpft in seinem Systementwurf daher konsequent präventive und reaktive Mechanismen. Die von ihm vorgeschlagenen Mechanismen zum lokalen und globalen Ausschluss auffälliger Knoten sind darüber hinaus ein erster Schritt in Richtung Angriffstoleranz und zur Durchsetzung semantischer Sicherheit.

Dieser letztgenannte Aspekt wird von den meisten anderen Lösungen kaum berücksichtigt. Einige Arbeiten sprechen das Problem zwar an, ohne jedoch konkrete Lösungsvorschläge zu unterbreiten. Die Konzentration der Schutzmaßnahmen auf das einzelne Endgerät und ihre Verbesserung durch Kooperation mit benachbarten Knoten muss angesichts der dynamischen Struktur mobiler Ad-hoc-Netze dennoch als grundsätzlich richtig angesehen werden. Die hierfür vorgeschlagenen Lösungen lassen jedoch in der Regel die Verlässlichkeit der Kooperationspartner weitestgehend außer acht. Dies mag damit zusammenhängen, dass sich semantische Sicherheit mit herkömmlichen Maßnahmen nicht hundertprozentig sicherstellen lässt. Byzantinisches Fehlverhalten einzelner Komponenten kann in offenen Umgebungen niemals ausgeschlossen werden und lässt sich auch nur schwer feststellen. Eine Möglichkeit, die hieraus resultierenden Auswirkungen dennoch zu mildern, stellen die in Kapitel 2.6 beschriebenen angriffstoleranten Ansätze für mobile Ad-hoc-Netze dar. Wie alle anderen Verfahren dieses Bereichs sind sie jedoch noch nicht praxiserprobt und gehen zudem oft von unrealistischen Annahmen über die Systemumgebung aus. Dennoch bilden sie einen wichtigen Baustein für eine umfassende Sicherheitslösung.

Der verteilt-kooperative Ansatz in mobilen Ad-hoc-Netzen steht derzeit noch im Widerspruch zu dem streng hierarchisch organisierten Systemmodell kommerzieller Wireless IDS, bei denen zwar die einzelnen Sensoren über das Netz verteilt sind, die Auswertung der gesammelten Informationen jedoch an zentraler Stelle erfolgt. Faktisch bauen heutige kommerzielle Wireless IDS auf herkömmlichen IDS-Architekturen auf, die lediglich um spezielle WLAN-Sensoren ergänzt wurden. Folglich kann von Dezentralisierung kaum die Rede sein. Die zunehmende Verbreitung von Policy Enforcement Tools lässt allerdings vermuten, dass das einzelne Endgerät in Zukunft immer mehr in den Mittelpunkt der Betrachtungen rücken wird. Hierfür spricht sicherlich auch, dass mit der wachsenden Verbreitung immer leistungsfähigerer mobiler Endgeräte samt

maßgeschneiderter Dienste immer weniger Nutzer die grundsätzliche Deaktivierung bestimmter Eigenschaften ihres Endgeräts hinnehmen werden. Dem ortsabhängigen Schutz des einzelnen Endgeräts wird also zwangsläufig mehr Aufmerksamkeit gewidmet werden müssen. Die ersten Arbeiten aus dem akademischen Bereich weisen hierbei in die richtige Richtung, wobei auch sie grundlegende Fragestellungen hinsichtlich Sicherheit und Verlässlichkeit weitestgehend unberücksichtigt lassen.

Mit zunehmender Gerätezentrierung wird zukünftig auch dem Aspekt Usability und Ergonomie eine größere Bedeutung zukommen. Bei der Gestaltung der Benutzerführung sind nicht nur die unterschiedlichen Eigenschaften der einzelnen Geräteklassen zu berücksichtigen, sondern es muss auch verstärkt auf den einzelnen Benutzer und seine Kenntnisse Rücksicht genommen werden. So mahnen aktuelle Tests bereits heute die Anbieter kommerzieller Systeme an, sich diesem Thema verstärkt zu widmen, obwohl diese Systeme von speziell geschultem Personal bedient werden [Bul06b]. Wie muss dann die Interaktion mit Nutzern konzipiert werden, denen grundlegendes Wissen zur Sicherheitsproblematik fehlt und die das Gerät primär für andere Aufgaben nutzen wollen?

Zusammenfassend ergibt sich damit die Forderung nach einer universellen Systemarchitektur zur Integration aller bestehenden Ansätze zum Schutz mobiler Umgebungen. Diese muss auf die besonderen Spezifika mobiler Ad-hoc-Netze als auch infrastruktureller Netze eingehen. Angesichts der dynamischen Struktur mobiler Ad-hoc-Netze muss der Architekturentwurf einen verteilt-kooperativen Ansatz verfolgen. Um dabei auch praxistauglich zu sein, ist der Sicherheit und Verlässlichkeit solcher Kooperationen besondere Aufmerksamkeit zu widmen.

| Kategorie | System bzw. Papier | Praxistauglichkeit | Anforderungen: funktional /R1/ Prävention | /R2/ Überwachung | /R3/ Reaktion | nichtfunktional /R4/ Skalierbarkeit | /R5/ Usability | /R6/ Sicherheit | /R7/ Verlässlichkeit | /R8/ Universalität | /R9/ Dezentralisierung |
|---|---|---|---|---|---|---|---|---|---|---|---|
| präventiv | *Architekturen für mobile Ad-hoc-Netze* | | | | | | | | | | |
| | Asymmetrische kryptographische | × | ✓ | × | × | − | ∅ | ∅ | − | − | + |
| | Symmetrische kryptographische | × | ✓ | × | × | ∅ | ∅ | ∅ | − | − | + |
| | Hybride kryptographische | × | ✓ | × | × | + | ∅ | ∅ | − | − | + |
| | *Architekturen für mobile infrastrukturelle Netze* | | | | | | | | | | |
| | WEP | × | ✓ | × | × | − | ∅ | − | − | − | − |
| | WPA | ✓ | ✓ | × | × | + | + | + | − | − | − |
| | RSN | ✓ | ✓ | × | × | + | + | + | − | − | − |
| reaktiv | *Architekturen für mobile Ad-hoc-Netze* | | | | | | | | | | |
| | [ZL00, ZLH03] | × | × | ✓ | × | ∅ | − | − | − | − | + |
| | LIDS [ACP+02] | × | × | ✓ | × | + | − | ∅ | ∅ | − | + |
| | [HL03] | × | × | ✓ | × | + | − | − | ∅ | − | + |
| | [PPM+03, PMPS04] | ✓ | × | ✓ | × | + | − | − | − | ∅ | + |
| | ZBIDS [SWP03] | × | × | ✓ | × | + | − | − | − | − | + |
| | SAM [Kar03] | × | ✓ | ✓ | ✓ | + | − | + | + | ∅ | + |
| | [YLY+04] | × | ✓ | ✓ | ✓ | ∅ | − | − | ∅ | ∅ | + |
| | [SBC+05] | × | × | ✓ | × | + | − | − | − | ∅ | + |
| | [DXL+06] | × | × | ✓ | × | + | − | − | ∅ | − | + |
| | *Architekturen für mobile infrastrukturelle Netze* | | | | | | | | | | |
| | DWSA [BNPDS04] | ✓ | × | ✓ | × | ∅ | ∅ | ∅ | − | − | − |
| | [SLO04] | ✓ | × | ✓ | ✓ | + | − | − | − | ∅ | + |
| | [ACL06] | × | × | ✓ | × | + | − | ∅ | ∅ | − | + |
| | WiFiProfiler [CPZ06] | ✓ | × | ✓ | ✓ | + | + | ∅ | ∅ | − | + |
| | kommerzielle Systeme | ✓ | ✓ | ✓ | ✓ | ∅ | ∅ | + | − | − | − |
| tolerant | Watchdog/Pathrater [MGLB00] | × | × | ✓ | ✓ | + | − | ∅ | + | ∅ | + |
| | CONFIDANT [BB02] | × | × | ✓ | ✓ | + | − | − | + | − | + |
| | CORE [MM02] | × | × | ✓ | ✓ | − | − | − | + | − | + |
| | OSRP [AHNRR02] | × | × | ✓ | ✓ | − | − | + | + | − | ∅ |
| | Nuglets [BH03] | × | × | × | × | + | − | + | + | − | ∅ |

✓ = erfüllt; × = nicht erfüllt; + = gut; ∅ = durchschnittlich; − = unzureichend

Tabelle 2.3: Vergleich von Sicherheitsarchitekturen für mobile Umgebungen

# 3
# Eine hybride universelle Sicherheitsarchitektur für mobile Systeme

Die Diskussion in Kapitel 2.7 nennt als grundlegende Schwäche quasi aller bisherigen Sicherheitsarchitekturen für mobile Umgebungen ihre eingeschränkte Wahrnehmung der Problematik. So werden nur Lösungen für einzelne Aspekte des Gesamtproblems vorgeschlagen, wobei die einzelnen Ansätze in der Regel auch noch von unrealistischen Grundannahmen ausgehen. Beispielweise postulieren viele der vorgestellten sicheren Ad-hoc-Routing-Protokolle die Existenz gegenseitiger geheimer Schlüssel ohne jedoch den hierfür notwendigen Schlüsselaustausch zwischen den beteiligten Knoten zu beschreiben. Für mobile Ad-hoc-Netze fordern daher bereits Kargl [Kar03] und Yang et al. [YLY+04] eine umfassende und aufeinander abgestimmte Sicherheitslösung. Während Yang et al. dabei nur abstrakt von einem widerstandsfähigen Architekturentwurf sprechen, macht Kargl mit SAM einen konkreten Realisierungsvorschlag. Dieser umfasst neben einem Ad-hoc-Routing-Protokoll Komponenten zur Erkennung von Angriffen sowie zur Identifizierung einzelner Netzknoten. In Ermangelung einer funktionsfähigen Testumgebung wurde SAM jedoch ausschließlich simulativ evaluiert.

Die im Folgenden vorgestellte *hybride universelle Sicherheitsarchitektur* (kurz HUSAR) verfolgt hingegen einen deutlich praxisorientierteren Ansatz. Erste Grundzüge von HUSAR wurden bereits in [Gro06a] und [Gro06b] umrissen. Ausgehend von den in Kapitel 2 ermittelten Sicherheitsanforderungen mobiler Umgebungen und der Analyse hierzu existierender Ansätze, soll HUSAR eine umfassende und in seinen Teilkomponenten aufeinander abgestimmte Sicherheitslösung für mobile Umgebungen realisieren. HUSAR legt eine Rahmenarchitektur zur Integration unterschiedlichster präventiver,

reaktiver und angriffstoleranter Sicherheitsmechanismen fest und stellt dabei den kooperativen Schutz des einzelnen Endgeräts in den Vordergrund. Im Gegensatz zu SAM erhebt der Ansatz außerdem den Anspruch auf Universalität, da sowohl Ad-hoc- wie auch Infrastruktur-Netze unterstützt werden. Außerdem spricht HUSAR ausdrücklich die semantische Sicherheit der Kooperationsbeziehungen an, indem bei der Auswahl von Kooperationspartnern deren bisheriges Verhalten als Maß für ihre Verlässlichkeit herangezogen wird.

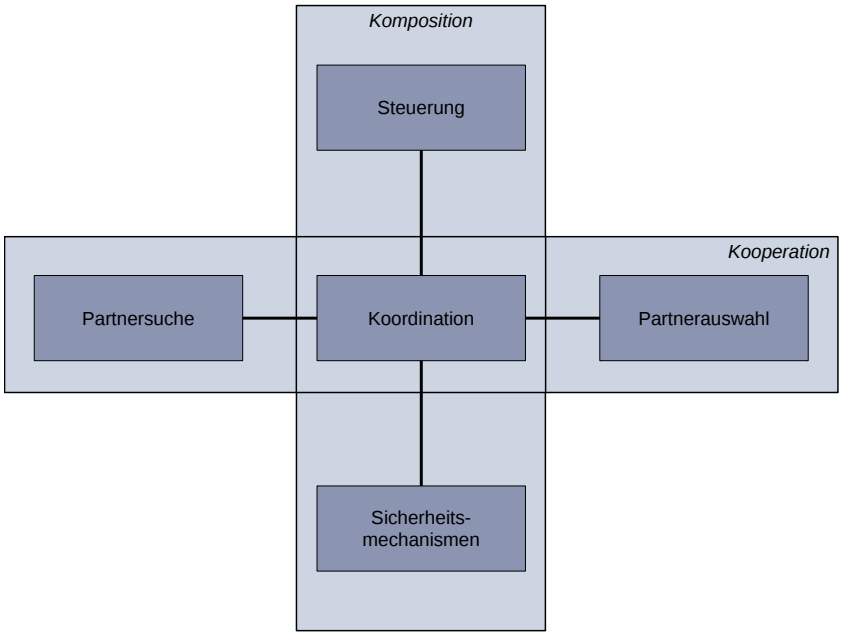



Abbildung 3.1: Übersicht der Funktionalität von HUSAR

Abbildung 3.1 ist eine Übersichtsdarstellung der von HUSAR realisierten Funktionalität. Diese fußt auf der als *Koordination* bezeichneten Verbindung einzelner Systeme und Systemteile. Dabei wird grundsätzlich zwischen der *Komposition* von Sicherheitsmechanismen zu einem umfassenden Gesamtsystem zum lokalen Schutz eines Endgeräts und der dazu orthogonalen *Kooperation* verschiedener unabhängiger Netzknoten im Rahmen einer verteilten Sicherheitslösung unterschieden. Die hierbei zu berücksichtigenden Aspekte werden im Folgenden näher erläutert.

## 3.1 Funktionale Bestandteile

Im folgenden werden die einzelnen Aspekte von HUSAR näher erläutert. Zunächst wird jeder funktionale Bestandteil anhand seiner Aufgaben genauer spezifiziert. Au-

ßerdem wird umrissen, welche der in Kapitel 2.3 auf Seite 23 gestellten Anforderungen erfüllt werden. Zum Abschluss werden die jeweils getroffenen Grundannahmen und notwendigen Voraussetzungen zusammengefasst.

### 3.1.1 Koordination

**Aufgabe:** Im Zentrum der vorgeschlagenen Architektur steht die enge Kopplung der auf einem mobilen Endgerät lokal implementierten Sicherheitsmechanismen, wie sie in den Kapiteln 2.4 bis 2.6 beschrieben wurden. Auf Grundlage des jeweiligen Anwendungsszenarios soll so ein umfassender Schutz vor möglichen Bedrohungen realisiert werden. Orthogonal hierzu wird der lose Zusammenschluss autonomer Netzknoten zu einem verteilt-kooperativen Verbund betrieben, um so den individuellen Schutz zu verbessern. Darüber hinaus wird eine einheitliche Schnittstelle zur Steuerung der einzelnen Komponenten bereitgestellt.

**Realisierte Anforderungen:** Die Komposition einer Sicherheitslösung aus einzelnen spezialisierten Sicherheitsmechanismen ermöglicht grundsätzlich die vollständige Abdeckung aller in Kapitel 2.3 beschriebenen funktionalen Anforderungen und trägt somit den dynamischen Aspekten der Netzsicherheit Rechnung (vgl. Abbildung 1.1 auf Seite 3). Im weiteren Verlauf konzentriert sich die vorliegende Arbeit allerdings auf die Frage der Überwachung und damit auf die Erkennung von Schutzzielverletzungen gemäß Anforderung */R2/*. Ferner erfüllt die Komposition eines universellen Gesamtsystems aus für bestimmte Netzzugangstechniken optimierten Sicherheitsmechanismen (Anforderung */R8/*). Dem Anspruch einer dezentral organisierten Lösung gemäß Anforderung */R9/* wird einerseits durch die Fokussierung auf das einzelne Endgerät, andererseits aber auch durch die lose Kopplung autonomer Netzknoten zu einem kooperierenden Verbund entsprochen. Die Bereitstellung einer einheitlichen Schnittstelle zur Steuerung der Gesamtlösung ist zudem ein indirekter Beitrag zur Verbesserung ihrer Gebrauchstauglichkeit (Anforderung */R5/*).

**Voraussetzungen und Annahmen:** Die enge Kopplung präventiver, reaktiver und angriffstoleranter Sicherheitsmechanismen zu einem hybriden Gesamtsystem setzt die permanente Verfügbarkeit der beteiligten Komponenten ebenso voraus wie die konsistente Beschreibung der hierfür notwendigen Schnittstellen. Beide Annahmen erscheinen im lokalen Umfeld als durchaus realistisch. Die Auswahl und Kombination geeigneter Sicherheitsmechanismen ist ferner von dem konkreten Anwendungsszenario abhängig. Für dieses müssen vorab die relevanten Umgebungsparameter und Sicherheitsanforderungen geklärt werden, was ein hohes Maß an Expertenwissen erfordert. Zudem wird postuliert, dass die Kombination sicherer Einzelmechanismen zu keinen neuen Gefährdungen für das Gesamtsystem führt.

### 3.1.2 Sicherheitsmechanismen

**Aufgabe:** Die einzelnen Sicherheitsmechanismen stellen die eigentliche Sicherheitsfunktionalität zur Verfügung, indem sie Teile einer Sicherheitsrichtlinie umsetzen (vgl. hierzu auch Kapitel 1.1). Jeder Mechanismus konzentriert sich dabei auf bestimmte Aspekte des Gesamtproblems. Wie in den vorangegangenen Kapiteln dargelegt wird hierzu einerseits zwischen präventiven, reaktiven und angriffstoleranten Mechanismen unterschieden, andererseits hinsichtlich der zugrunde liegenden Systemumgebung.

**Realisierte Anforderungen:** Je nach Ausprägung erfüllt ein konkreter Sicherheitsmechanismus eine oder mehrere der aufgestellten funktionalen Anforderungen. Auch die Abdeckung der nichtfunktionalen Anforderungen ist von der jeweiligen Umsetzung des Mechanismus abhängig. Eine detaillierte Übersicht existierender Sicherheitsmechanismen und der von ihnen umgesetzten Anforderungen wurde bereits in den Kapiteln 2.4, 2.5 und 2.6 gegeben.

**Voraussetzungen und Annahmen:** Um einzelne Mechanismen miteinander koppeln zu können, muss im wesentlichen ihre Schnittstelle klar beschrieben sein. Hierzu gehört auch die Ermittlung notwendiger Voraussetzungen für den Einsatz eines Mechanismus sowie seine Zielsetzung. Dies setzt, wie bereits auf der vorherigen Seite angesprochen, fundierte Kenntnisse voraus.

### 3.1.3 Partnersuche und -auswahl

**Aufgabe:** Neben der Kombination lokal vorhandener Sicherheitsmechanismen trägt auch die Kooperation mit anderen Netzknoten zur Verbesserung der individuellen Sicherheit bei. Wie Abbildung 3.2 darstellt, kommen hierbei zwei Effekte zum Tragen.

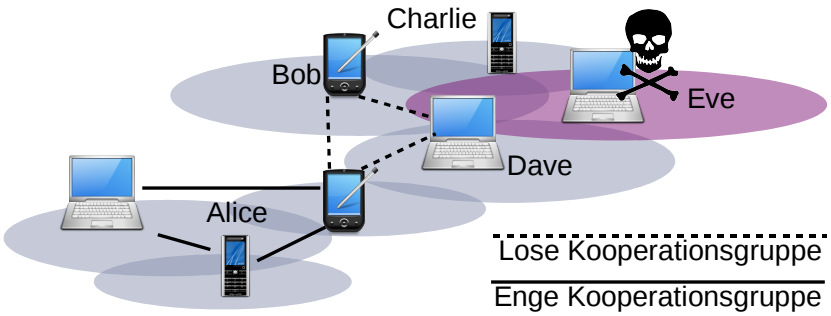



Abbildung 3.2: Verbesserung der individuellen Sicherheit durch Kooperation

Zum einen wird das frühzeitige Erkennen von Bedrohungen ermöglicht. So wird Alice von Bob und Dave bereits vor der von Eve ausgehenden Gefährdung gewarnt, bevor sie überhaupt in deren Wirkungskreis gerät. Zum anderen können mobile Geräte gemeinsam ihre knappen Ressourcen effizienter nutzen.

Um Kooperationen eingehen zu können, müssen zunächst potentielle Partner gesucht und gefunden werden, wozu erst einmal eine Möglichkeit zur Kommunikation geschaffen werden muss. Da eine Interaktion mit anderen Geräten immer zu Lasten der vorhandenen Ressourcen geht, sollte bei der Auswahl von Partnern der erwartete Nutzen mit den entstehenden Kosten abgewogen werden. Besonderes Gewicht hat hierbei auch das in einen Kooperationspartner gesetzte Vertrauen. In herkömmlichen Sicherheitsarchitekturen werden solche Vertrauensbeziehungen häufig durch den Austausch von Geheimnissen und im Rahmen kryptographischer Protokolle realisiert. Ähnliche Verfahren wurden auch für mobile Umgebungen vorgeschlagen. So könnte Alice beispielsweise den in [SA02] beschriebenen Ansatz des *Resurrecting Duckling* verwenden, um ihren Laptop, PDA und ihr Smartphone zu einer engen Kooperationsgruppe zusammenzuschließen. Als *Kooperationsgruppe* wird dabei eine Menge gemeinsam auf ein Ziel hin arbeitender mobiler Endgeräte bezeichnet [Gro06a, Gro06b]. Herrscht vollkommenes gegenseitiges Vertrauen zwischen den einzelnen Endgeräten, teilen sie nicht nur ihr Wissen, sondern auch ihre Systemressourcen. Ein solcher Zusammenschluss wird *enge Kooperationsgruppe* genannt. Mit wachsender Dynamik der Netzumgebung skaliert dieser Ansatz jedoch immer schlechter. Alternativ können daher *lose Kooperationsgruppen* gebildet werden, in denen die beteiligten Endgeräte einander nur teilweise trauen. Sie tauschen deshalb nur gegenseitig Informationen aus, gewähren darüber hinaus jedoch keinen Zugriff auf ihre Systemressourcen. Um möglichem Missbrauch in losen Kooperationsgruppen vorzubeugen, wurden bereits in Kapitel 2.6 sogenannte *Soft Security* Mechanismen vorgestellt.

**Realisierte Anforderungen:** Die gemeinsame Nutzung knapper Ressourcen zielt auf die Verbesserung der Skalierbarkeit der Sicherheitslösung (Anforderung */R4/*). Zudem verbessert die oben beschriebene Antizipation von Angriffen die Sicherheit (Anforderung */R6/*). Der Einsatz von *Soft Security* Mechanismen dient der Verbesserung der Verlässlichkeit kooperierender Knoten (Anforderung */R7/*). Die Sanktionierung unbotmäßigen Verhaltens durch hierfür verwendbare Reputationssysteme kann schließlich als Umsetzung der Anforderung */R3/* gewertet werden.

**Voraussetzungen und Annahmen:** Grundlegende Voraussetzung der Suche und Auswahl von Partnern ist deren eindeutige Identifizierbarkeit. Die Verwendung von *Soft Security* Mechanismen, wie etwa Reputationssystemen zur Auswahl zuverlässiger Partner, konkretisieren diese Anforderung weiter, indem sie langlebige Identitäten erfordern.

### 3.1.4 Steuerung

**Aufgabe:** Die Separation der Steuerung von dem eigentlichen System ermöglicht die Realisierung einer adaptiven Nutzerschnittstelle, die sich sowohl an die Bedürfnisse des jeweiligen Benutzers anpasst als auch an die technischen Gegebenheiten des genutzten Endgeräts.

**Realisierte Anforderungen:** Dieser Aspekt von HUSAR zielt also im wesentlichen auf die Gebrauchstauglichkeit (Anforderung */R5/*), aber auch auf die Verwendung mit beliebigen Endgeräten, selbst solchen mit keinen oder nur eingeschränkten Möglichkeiten zur Interaktion mit dem Benutzer.

**Voraussetzungen und Annahmen:** Wesentliche Voraussetzung einer entkoppelten Steuerung ist die genaue Spezifikation ihrer Schnittstelle zur Koordinationskomponente sowie jener zwischen der Koordinationskomponente und den einzelnen Sicherheitsmechanismen, um deren Funktionalität umfassend zu kapseln.

## 3.2 Verallgemeinerung des gewählten Ansatzes

Bevor es an die konkrete Ausgestaltung des vorgestellten Architekturentwurfs geht, soll zunächst noch eine mögliche Verallgemeinerung des Konzepts angesprochen werden.

Für einzelne Sicherheitsmechanismen – im Folgenden wird der konkrete Fall der Angriffserkennung betrachtet – sieht HUSAR derzeit die horizontale Kopplung autonom agierender Endgeräte zu kooperierenden Gruppen vor. Dieser Kopplungsprozess zur Laufzeit trägt der hohen Dynamik mobiler Ad-hoc-Netze Rechnung. Die vertikale Komposition der realisierten Sicherheitslösung aus einzelnen Sicherheitsmechanismen wird jedoch durch den Entwickler statisch vorgegeben.

Als logischer nächster Schritt drängt sich nicht nur die Zusammenarbeit einzelner Endgeräte bei unterschiedlichen Sicherheitsmechanismen, sondern auch deren dynamische Komposition geradezu auf. Eine solche kooperative Komposition von Sicherheitsmechanismen würde eine im weitesten Sinne dienstorientierte Architektur realisieren, wie sie in Abbildung 3.3 auf der nächsten Seite dargestellt ist.
Ein *Sicherheitsdienst* erweitert einen herkömmlichen Sicherheitsmechanismus dabei um die formale Spezifikation seiner Eigenschaften. Hierzu gehören beispielsweise die bereitgestellten Schnittstellen, benötigte Voraussetzungen sowie die realisierten Sicherheitseigenschaften des Mechanismus, aber auch die Identität seines Anbieters.

Das so entstehende System würde sich deutlich besser den sich verändernden Umgebungsbedingungen anpassen. Beispielsweise könnten abhängig von der momentan genutzten Netzzugangstechnologie die verwendeten Sicherheitsmechanismen automa-

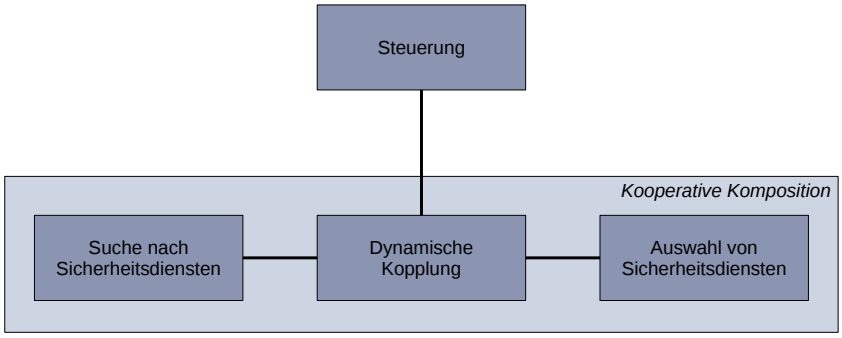



Abbildung 3.3: Dynamische Komposition verteilter Sicherheitsdienste

tisch rekonfiguriert werden, um so der Anforderung */R8/ Universalität* stärker Rechnung zu tragen.

Ähnlich könnte auf die Verknappung von Ressourcen reagiert werden, indem beispielsweise weniger sichere, dafür aber ressourcensparsamere Mechanismen ausgewählt würden. Diese dynamische Kalkulation des Trade-off zwischen Sicherheit und Funktionalität ist einer besseren Berücksichtigung der Anforderungen */R4/ Skalierbarkeit* und */R6/ Sicherheit* gleichzusetzen.

Als wesentliche Voraussetzung für diesen erweiterten Ansatz bedarf es allerdings einer genauen Spezifikation nicht nur der Kooperationspartner, sondern auch der Sicherheitsmechanismen, um diese lokalisieren und einbinden zu können. Während ersteres noch relativ einfach zu realisieren ist, erfordert letzteres eine sehr klare Modellbildung und die genaue Erfassung der geforderten Sicherheitseigenschaften sowie der aktuellen Systemumgebung. Beides sind Aufgaben, deren praktische Umsetzung den Rahmen dieser Arbeit bei weitem sprengt. Im Folgenden wird daher dem ursprünglichen Ansatz gefolgt.

## 3.3 Zusammenfassung

Damit ist die grundlegende Funktionalität von HUSAR klar umrissen. Ihre genaue Ausgestaltung ist Gegenstand der nun folgenden Kapitel, wobei sich die vorliegende Arbeit auf die kooperative Erkennung von Angriffen konzentriert. Anhand der in Kapitel 2 eingeführten Kriterien kann die Architektur wie in Tabelle 3.1 zusammengefasst bewertet werden.
Die im weiteren Verlauf beschriebene prototypische Realisierung von HUSAR basiert auf dem ursprünglichen Architekturmodell. Dementsprechend wird in Tabelle 3.1 die Praxistauglichkeit des verallgemeinerten Modells verneint.

| Kategorie | System | Praxistauglichkeit | Anforderungen | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | funktional | | | nichtfunktional | | | | | |
| | | | /R1/ Prävention | /R2/ Überwachung | /R3/ Reaktion | /R4/ Skalierbarkeit | /R5/ Usability | /R6/ Sicherheit | /R7/ Verlässlichkeit | /R8/ Universalität | /R9/ Dezentralisierung |
| hybrid | HUSAR gemäß Abschnitt 3.1 | ✓ | ✓ | ✓ | ✓ | ∅ | ∅ | ∅ | + | ∅ | + |
| | HUSAR gemäß Abschnitt 3.2 | × | ✓ | ✓ | ✓ | + | + | + | + | + | + |

✓ = erfüllt; × = nicht erfüllt; + = gut; ∅ = durchschnittlich; − = unzureichend

Tabelle 3.1: Zusammenfassende Bewertung von HUSAR

Kapitel 4 behandelt im Folgenden die prototypische Implementierung einer Systemlösung zur Unterstützung der Angriffserkennung in Wireless LANs. Diese folgt zunächst einem lokalen Ansatz, um dann in einem zweiten Schritt zu einer verteilten Lösung erweitert zu werden.

Die notwendige Funktionalität zur Verwaltung der hierbei kooperierenden Netzknoten ist Gegenstand von Kapitel 5. Dieses beschreibt am Beispiel des bereits angesprochenen Wireless IDS die Einbindung lokaler Sicherheitsmechanismen in die Rahmenarchitektur sowie die lose Kopplung autonom agierender mobiler Knoten zu einem kooperierenden Verbund. Wesentlicher Teil des hierbei entwickelten Prototypen ist die Realisierung einer Infrastruktur zur Suche nach potentiellen Kooperationspartnern auf Basis von Peer-to-Peer-Mechanismen.

# 4

# Systemlösung zur Angriffserkennung in Wireless LANs

Als erster Teil der Realisierung des soeben beschriebenen Architekturentwurfs wird nun ein verteiltes Network Intrusion Detection System für Wireless LANs präsentiert. Dessen prototypische Umsetzung erfolgte im Rahmen einer Diplom- [Neu05] und einer Studienarbeit [Rei06]. Die hierbei erzielten Ergebnisse wurden darüber hinaus in [GN06] vorgestellt.

Als technische Basis dient das quelltextoffene IDS-Framework Bro[1], das jedoch um spezielle Mechanismen für die Verarbeitung von Paketen gemäß des IEEE-Standards 802.11 sowie um maßgeschneiderte Strategien zur Erkennung von Angriffen in solchen Netzen erweitert wurde.

Die Wahl von Bro im Rahmen einer ersten prototypischen Umsetzung der vorgeschlagenen Sicherheitsarchitektur wird dabei wie folgt begründet: Zum einen stellt Bro ein weitreichendes Arsenal an Werkzeugen zur Verfügung, die vom verwendeten Übertragungsprotokoll abstrahierend eine universelle Erfassung und Analyse übermittelter Netzwerkpakete ermöglichen. Dabei werden sowohl anomalie- als auch missbrauchserkennende Ansätze unterstützt. Zum anderen wurde Bro für den Einsatz in breitbandigen Hochgeschwindigkeitsnetzen und damit mit besonderem Augenmerk auf Ressourceneffizienz entwickelt, was eine Verwendung auf ressourcenknappen mobilen Endgeräten begünstigen sollte. Schließlich unterstützt Bro bereits von Haus aus die verteilte Erkennung von Angriffen.

[1] http://www.bro-ids.org/ (Abruf: August 2008)

Im Folgenden wird die durchgeführte Systemimplementierung zunächst konzeptuell vorgestellt. Dabei werden die notwendigen Systemerweiterungen dem in Kapitel 2.5.1 eingeführten allgemeinen IDS-Architekturmodell zugeordnet. Nach einer kurzen Einführung in Bro wird dann die konkrete Umsetzung im Detail beschrieben. Neben Anpassungen des Bro Kerns gehört hierzu insbesondere die Entwicklung neuer Policies für mobile Systeme. Für konkrete Angriffe werden zunächst allgemeine Strategien für ihre Erkennung beschrieben, woraus dann spezifische Bro Policies abgeleitet werden. Abschließend wird die Validierung des so entstandenen Prototypen im Rahmen mehrerer Laborversuche erläutert.

## 4.1 Konzeption

Abbildung 4.1 fasst die für die Unterstützung mobiler Umgebungen notwendigen Erweiterungen an der in Kapitel 2.5.1 vorgestellten allgemeinen IDS-Architektur zusammen. Diese erstrecken sich über die gesamte Systemarchitektur, ausgehend von der Datenerfassung über ihre Analyse und Interpretation bis hin zur Reaktion auf die so erkannten Angriffe. Die einzelnen Anpassungen leiten sich dabei direkt von dem konkret verwendeten Übertragungsprotokoll ab, erfordern somit eine systematische Untersuchung seiner Eigenheiten.

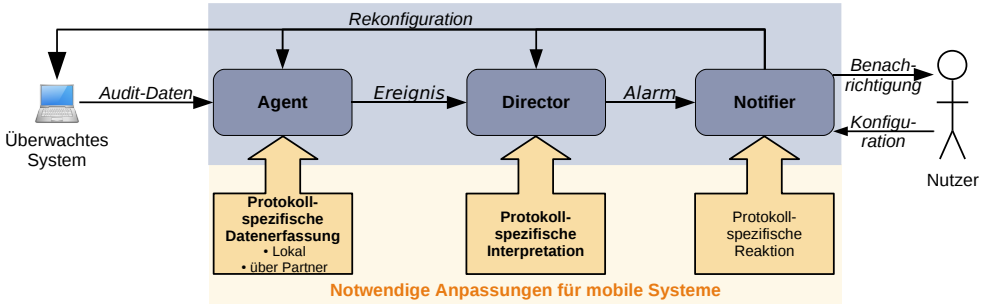



Abbildung 4.1: Erweiterung der allgemeinen IDS-Architektur für mobile Systeme

Um der in Kapitel 1.2 gestellten Forderung nach fundamentalem Schutz nachzukommen, sind alle drei logischen Komponenten eines IDS auf jedem mobilen Endgerät zu implementieren. Der Forderung nach kooperativem Schutz zur Verbesserung der individuellen Erkennungsrate wird dadurch Rechnung getragen, dass neben den lokal erhobenen Auditdaten auch von anderen Netzknoten übermittelte Informationen berücksichtigt werden.

Die Vorgehensweise zur Entwicklung einer Angriffserkennung für mobile Netze lässt sich damit wie folgt umreißen:

*Protokollspezifische Datenerfassung:* Hierfür ist zunächst zu klären, welche Daten zur Erkennung auffälliger Systemzustände relevant sind. Die Identifikation dieser charakteristischen Merkmale erfolgt dabei auf Grundlage einer sorgfältigen Analyse der Spezifikation des Übertragungsprotokolls, beruht aber auch auf der Betrachtung bereits bekannter Angriffe. Darüber hinaus müssen die technischen Gegebenheiten zum Erheben dieser Daten geschaffen werden. Dies umfasst neben dem Zugriff auf die Rohdaten der Netzwerkschnittstelle auch den Informationsaustausch mit kooperierenden Knoten. Schließlich müssen die so gewonnenen Informationen zu abstrakten Ereignissen transformiert werden, auf deren Grundlage die Einschätzung des aktuellen Systemzustands durch den *Director* erfolgt.

*Protokollspezifische Dateninterpretation:* Ebenso wie die Erfassung der Daten erfordert auch ihre Interpretation den Entwurf eines Systemmodells auf Grundlage der Spezifikation des Übertragungsprotokolls und der Analyse bekannter Angriffe. Auf diese Weise lassen sich sowohl anomalie- als auch missbrauchserkennende Strategien umsetzen.

*Protokollspezifische Reaktion:* Auch die Planung und Gestaltung von Reaktionsmechanismen auf erkannte Angriffe ist abhängig von der betrachteten Systemumgebung [Bac00b]. Sie wird damit ebenfalls durch das verwendete Übertragungsprotokoll beeinflusst, muss darüber hinaus aber noch eine Reihe weiterer der in Kapitel 2.3 genannten Einflussfaktoren berücksichtigen. Von besonderer Bedeutung sind hierbei die Nutzerzentrierung mobiler Endgeräte und ihre Nutzerschnittstelle. So müssen die Auswirkungen entdeckter Angriffe und möglicher Reaktionen hierauf für den Nutzer nachvollziehbar sein. Der *Notifier* muss dazu individuell zugeschnittene Hilfestellungen anbieten, die auch die möglichen Einschränkungen hinsichtlich der vom Endgerät angebotenen Nutzerschnittstelle berücksichtigen. Diese Anforderungen wurden bereits in Kapitel 2.3.3 unter Punkt */R5/* zusammengefasst.

Die bei der Gestaltung der Interaktion von Sicherheitssystemen mit dem Nutzer auftretenden Probleme analysiert Whitten in [WT99, Whi04]. Ausgehend von fünf charakteristischen Eigenschaften, in denen sich Sicherheitssysteme von herkömmlichen unterscheiden, etabliert sie eine Reihe von Prinzipien für die Entwicklung gebrauchstauglicher Systeme. Diese unterstützen den Entwickler bei der Entscheidung, wann Sicherheitsmechanismen automatisiert und vom Nutzer unbemerkt durchgeführt werden können beziehungsweise wann eine Interaktion mit diesem notwendig ist. Ihre Erkenntnisse validiert sie experimentell am Beispiel einer E-Mail-Anwendung mit Public-Key-Verschlüsselung. Die beiden von ihr entwickelten Entwurfstechniken *Safe Staging* und *Metaphor Tailoring* eignen sich jedoch nicht für das in dieser Arbeit betrachtete Anwendungsfeld. Entweder negieren sie die Notwendigkeit zeitlich naher Reaktionen auf einen Angriff oder aber sie

setzen das Vorhandensein einer visuellen Nutzerschnittstelle voraus. Dennoch verdeutlicht Whittens Arbeit die vielfältigen Schwierigkeiten bei der Entwicklung gebrauchstauglicher Sicherheitssysteme, die nicht nur technischer Natur sind, sondern zum Teil auch Bereiche der Psychologie, Soziologie und Ökonomie berühren.

Eine detaillierte Behandlung von Reaktionsmechanismen für Angriffe wird daher im weiteren Verlauf weitestgehend ausgeblendet, um den Rahmen der Arbeit nicht zu sprengen. So ist der entwickelte Prototyp zwar modular aufgebaut und damit offen für die Erweiterung um aktive Gegenmaßnahmen. In seiner jetzigen Form unterstützt er jedoch ausschließlich die Protokollierung erkannter Angriffe.

## 4.2 Das Intrusion Detection Framework Bro

Bro [Pax99] ist ein flexibles Framework zur Erstellung von NIDS, dessen Entwicklung maßgeblich von einem Team um Vern Paxson mit Förderung der National Science Foundation vorangetrieben wird. Die wesentlichen Ziele sind hierbei

- die strikte Trennung der standortspezifischen Sicherheitsrichtlinien von den Mechanismen zur Erfassung und Analyse der Audit-Daten,
- Resistenz gegen Angriffe auf das IDS selbst sowie
- die effiziente Nutzung der zur Verfügung stehenden Ressourcen.

Dies schlägt sich in einer zweigeteilten Systemarchitektur nieder, die in Abbildung 4.2 auf der nächsten Seite dargestellt wird. Im folgenden werden die einzelnen Bestandteile von Bro kurz vorgestellt.

### 4.2.1 Der Bro Kern

Im Bro Kern werden sämtliche Mechanismen zur Erhebung von Audit-Daten, ihrer Analyse und der daraus resultierenden Generierung policy-neutraler Ereignisse zusammengefasst. Policy-neutral bedeutet, dass zunächst keine Bewertung ob des Eintretens eines bestimmten Ereignisses getroffen wird. Die Interpretation hiervon wird vielmehr durch die auf *Policy-Ebene* definierten Richtlinien festgelegt. Die einzelnen Bestandteile des in C++ implementierten Bro Kerns werden im Folgenden näher erläutert.

#### Network Analysis

Bro analysiert den Netzverkehr der Schichten 3 bis 7 des OSI-Referenzmodells. Es verwendet zunächst die Bibliothek libpcap[2], um den Netzverkehr an beliebigen Netz-

[2] http://sourceforge.net/projects/libpcap/ (Abruf: August 2008)

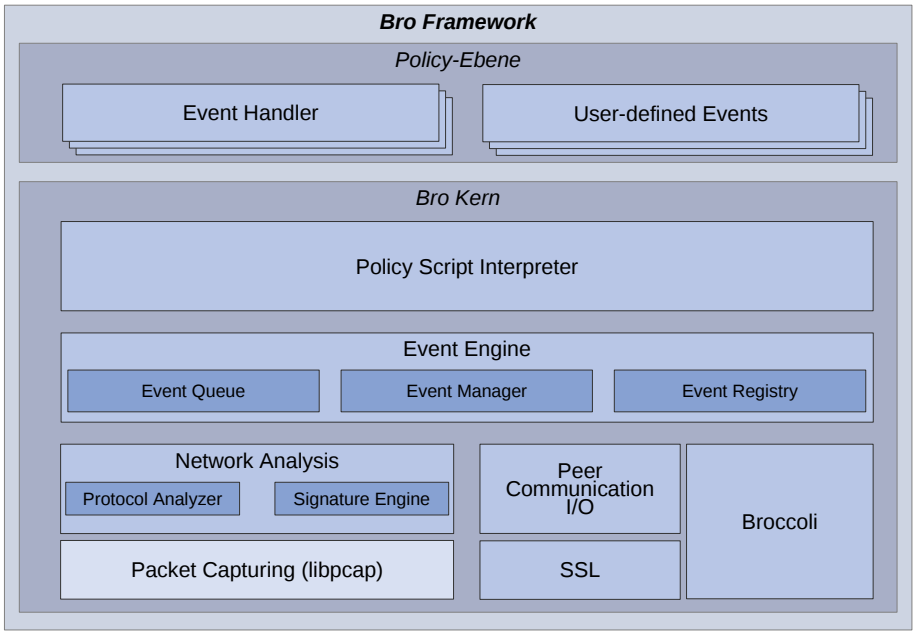



Abbildung 4.2: Architektur von Bro nach [KS05]

werkschnittstellen mitzulesen. Im Anschluss kommen zwei Techniken zur Klassifikation des erhobenen Netzverkehrs zum Einsatz. Eine *Signature Engine* durchsucht den eingehenden Datenstrom nach Mustern bekannter Angriffe, sogenannten Signaturen. Neben dieser Missbrauchserkennung unterstützt Bro mit Hilfe der *Protocol Analyzer* auch eine anomalieerkennende Strategie zur Feststellung von Angriffen. Die *Protocol Analyzer* beschreiben den korrekten Verlauf eingesetzter Netzwerk- und Anwendungsprotokolle, wie etwa ICMP, UDP, TCP, HTTP, SMB oder auch des P2P-Systems Gnutella. Sowohl die Durchführung bestimmter Aktivitäten als auch Abweichungen vom normalen Protokollablauf können dabei ebenso zur Generierung von Ereignismeldungen führen wie die Erkennung von Angriffsmustern durch die *Signature Engine*. Die so erzeugten Ereignisse reflektieren die Netzwerkaktivität und werden als policy-neutral bezeichnet, treffen also noch keine Aussage hinsichtlich ihrer Bedeutung.

#### Peer Communication und Broccoli

Neben der Analyse des lokal mitgeschnittenen Netzverkehrs können auch Informationen entfernter Bro-Knoten oder lokaler Programme als Quelle von Audit-Daten herangezogen werden. Hierfür steht im Bro Kern die Komponente *Peer Communication I/O* bereit, die Netzwerkverbindungen auf Basis von Sockets bereitstellt. Sie erlaubt

die Übermittlung beliebiger Ereignismeldungen zwischen einzelnen Bro-Knoten. Die Kommunikation kann dabei mittels SSL/TLS geschützt werden.

Darüber hinaus existiert mit *Broccoli*[3] eine weitere Schnittstelle, um lokale Applikationen an Bro anzubinden. *Broccoli* arbeitet dazu als Mittler zwischen diesen Anwendungen und dem Bro-Kommunikationsprotokoll. Die Anwendungen können damit Ereignismeldungen kreieren, senden, anfragen und empfangen sowie mit anderen Bro-Knoten interagieren. Dies lässt sich beispielsweise dazu nutzen, um die Erkennungsleistung von Bro mit host-spezifischen Kontextinformationen zu verbessern [DKPS05]. Weitere Anwendungsmöglichkeiten ergeben sich bei der Konfiguration und Verwaltung von Bro-Knoten oder auch in der Zusammenarbeit mit anderen Systemen.

#### Event Engine

Die *Event Engine* verwaltet die erzeugten policy-neutralen Ereignisse und koordiniert deren Verarbeitung und Interpretation. Hierzu werden Verweise auf die *Event Handler* der *Policy-Ebene* und die von ihnen verarbeiteten Ereignistypen zunächst in der *Event Registry* eingetragen. Ferner werden alle neu eingehenden Ereignismeldungen in einer *Event Queue* gepuffert. Der *Event Manager* koordiniert schließlich die Verknüpfung der *Queue* mit der *Registry*, indem er die Ereignisse der *Event Queue* nach dem FIFO-Prinzip zur weiteren Bearbeitung an die entsprechenden *Event Handler* übergibt. Hierzu überprüft er, ob bei der Analyse eines Pakets durch die *Network Analysis* Ereignismeldungen angefallen sind. Falls ja, arbeitet er diese zunächst der Reihe nach ab, bevor die *Network Analysis* mit der Untersuchung des nächsten eingehenden Pakets fortfährt.

#### Policy Script Interpreter

Der *Policy Script Interpreter* führt die auf *Policy-Ebene* definierten Policy-Skripte aus und stellt damit das Bindeglied zwischen der *Policy-Ebene* und dem Bro Kern dar. Aus Sicht der *Policy-Ebene* legt der Interpreter dabei den Umfang der Skriptsprache zur Definition der Sicherheitsrichtlinien fest. Aus Sicht des Bro Kerns sorgt er für die Interpretation der policy-neutralen Ereignisse, indem er die Anweisungen der Policy-Skripte deutet und auf entsprechende Funktionen des Kerns abbildet.

Beim Start sorgt der Interpreter zunächst dafür, dass alle aktiven *Event Handler* in der *Event Engine* registriert werden. Im weiteren Verlauf führt er für jedes vom *Event Manager* übergebene Ereignis die entsprechenden *Event Handler* aus, wobei er die Werte des Ereignisses den entsprechenden Argumenten des *Handlers* zuordnet. Ein *Event Handler* kann dabei beliebige weitere Skriptbefehle ausführen, um neue Ereignisse zu generieren, weitere *Event Handler* aufzurufen, den internen Zustand des IDS anzupassen, den Verlauf der Bearbeitung zu protokollieren oder auch um Reaktionen auf einen erkannten Angriff einzuleiten.

[3] http://www.icir.org/christian/broccoli/ (Abruf: August 2008)

### 4.2.2 Die Bro Policy-Ebene

Im Gegensatz zu den in C++ implementierten Komponenten des Bro Kerns werden die Policy-Skripte der *Policy-Ebene* in einer speziellen prozeduralen Skriptsprache, der Bro-Sprache, realisiert. Ihre Syntax ist an C angelehnt. Die Skripte legen die lokale Sicherheitsrichtlinie eines Systems fest und geben damit den aufgetretenen policy-neutralen Ereignissen eine Bedeutung.

Die Bro-Sprache ist streng getypt und bietet neben gängigen Datentypen und Operatoren auch speziell auf das Anwendungsgebiet zugeschnittene Sprachkonstrukte wie etwa Datentypen für IP-Adressen, Hostnamen oder Ports. Anweisungen lassen sich zu Funktionen zusammenfassen. Ferner wird die Verwendung regulärer Ausdrücke sowie lokal oder global gültiger Variablen und Konstanten unterstützt. Eine umfassende Darstellung der Bro Skriptsprache bietet [Pax99, Kapitel 3].

Die Beschreibung der lokalen Sicherheitsrichtlinie erfolgt mit Hilfe der *Event Handler*. Sie legen fest, wie auf bestimmte Ereignisse reagiert wird. Darüber hinaus können weitere Ereignisse definiert werden und via *Policy Script Interpreter* und *Event Manager* der *Event Queue* hinzugefügt werden.

### 4.2.3 Übersicht über die Implementierung

Nach dieser allgemein gehaltenen Darstellung der Bro-Architektur wird nun ihre prinzipielle implementierungstechnische Realisierung zusammengefasst, um auf die im Folgenden beschriebenen Erweiterungen für WLANs vorzubereiten.

Abbildung 4.3 auf der nächsten Seite fasst dazu die wesentlichen Aspekte der Objektmodellierung des Bro Kerns zusammen.

Die Funktionalität der *Network Analysis* ist im wesentlichen in vier Klassen gekapselt. PktSrc implementiert die Schnittstelle zu den Funktionen der C-Bibliothek libpcap. Die so erfassten Datenpakete werden durch die Klasse NetSessions zunächst einer Vorverarbeitung unterzogen. Hierzu gehören diverse Konformitätstests, die Einordnung in einen zeitlichen Übertragungskontext und, falls nötig, ihre Defragmentierung. Darüber hinaus werden der IP-Header zur weiteren Analyse extrahiert und die Header der Schichten 1 und 2 verworfen.

Die weitere Verarbeitung erfolgt dann durch eine Unterklasse der Klasse Connection, die somit einen konkreten *Protocol Analyzer* implementiert. Alternativ sucht die Klasse RuleMatcher nach bekannten Angriffsmustern und realisiert damit die Funktionalität der oben beschriebenen *Signature Engine*.

Sowohl Connection als auch RuleMatcher generieren policy-unabhängige Ereignisse, die in der durch die Klasse Event realisierten Warteschlange gepuffert und mit Hilfe der in der Klasse EventRegistry registrierten *Event Handler* der *Policy-Ebene* interpretiert werden. Die Koordination dieses Vorgangs wird durch die Klasse EventMgr gesteuert, der

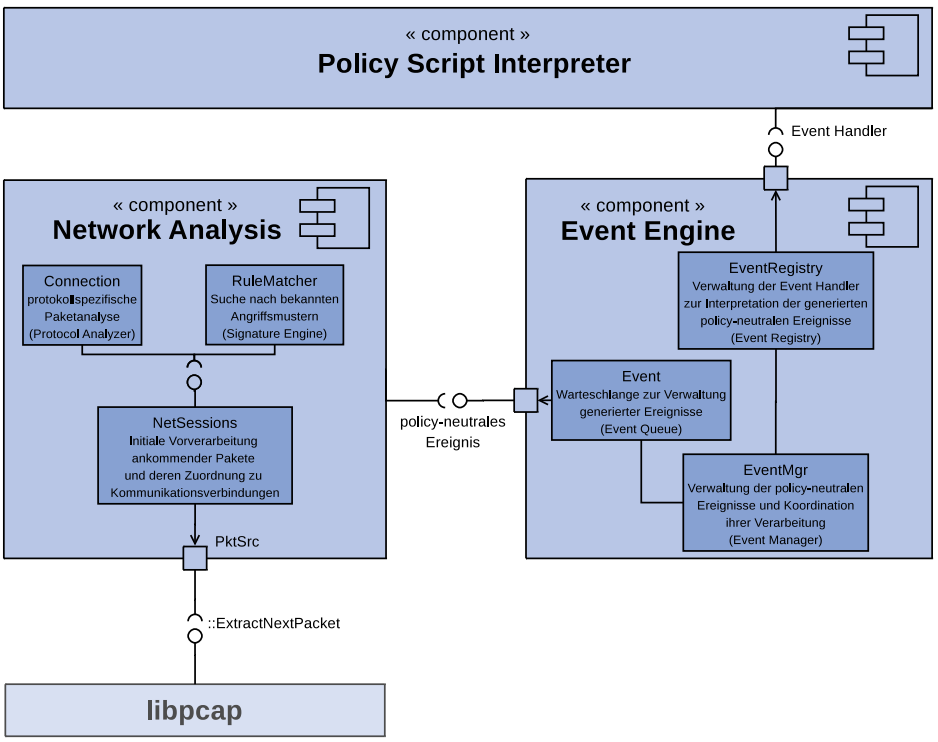



Abbildung 4.3: Allgemeines Objektmodell des Bro Kerns

insbesondere auch die hierfür notwendige Interaktion mit dem *Policy Script Interpreter* koordiniert.

### 4.2.4 Einordnung in das allgemeine Architekturmodell

Zur weiteren Vorbereitung der vorgenommenen Erweiterungen wird die Bro-Architektur wie folgt auf die allgemeine IDS-Architektur abgebildet und so mit dem in Kapitel 4.1 dargelegten Konzept für die Entwicklung eines IDS für mobile Umgebungen in Bezug gebracht:

*Agent:* Die untere Schicht des Bro Kerns, bestehend aus *Packet Capturing*, *Network Analysis*, *Peer Communication I/O* und *Broccoli*, bildet den *Agent* zur Erfassung der Auditdaten.

*Director:* Der für die Interpretation der vom *Agent* generierten Ereignisse zuständige *Director* setzt sich aus dem oberen Teil des Bro Kerns und Teilen des Policy Layers zusammen. Neben der *Event Engine* und dem *Policy Script Interpreter* umfasst er damit Teile der *Event Handler* und der nutzerdefinierten Ereignisse.

*Notifier:* Die Definition von Maßnahmen gegen erkannte Angriffe als auch die Entscheidung über deren Einleitung erfolgt ebenfalls in den *Event Handlern*. Somit ist der *Notifier* der Policy-Schicht zuzurechnen.

Wie bereits in Kapitel 4.1 erläutert, konzentriert sich der vorliegende Prototyp auf die beiden erstgenannten Systemkomponenten, nämlich den *Agent* und den *Director*. Damit fasst Tabelle 4.1 die allgemein notwendigen Erweiterungen zusammen, um Bro in mobilen Umgebungen einzusetzen.

| Allgemeine Architekturkomponente | Konkrete Bro-Komponente | Notwendige Erweiterungen |
|---|---|---|
| Agent | Packet Capturing | Übertragungsprotokollspezifische Erfassung lokal empfangenen Netzverkehrs |
| | Network Analysis | Übertragungsprotokollspezifische Aggregation lokal erfasster Verkehrsdaten zu policy-neutralen Ereignissen, Erkennung spezifischer Angriffssignaturen |
| Director | Event Engine | Definition von übertragungsprotokollspezifischen policy-neutralen Ereignissen |
| | Policy Script Interpreter | Definition übertragungsprotokollspezifischer Sprachkonstrukte |
| | Policy Layer | Übertragungsprotokollspezifische Interpretation der policy-neutralen Ereignisse |

Tabelle 4.1: Adaption von Bro an mobile Umgebungen

Hierfür ist zunächst ein Zugang zur konkret verwendeten Netzwerkschnittstelle und damit zu dem lokal empfangbaren Netzverkehr zu schaffen. Die so erlangten Daten sind dann entsprechend des verwendeten Übertragungsprotokolls zu analysieren und zu spezifischen policy-neutralen Ereignissen zusammenzufassen. Außerdem besteht die Möglichkeit, die *Signature Engine* um übertragungsprotokollspezifische Angriffsmuster zu erweitern und so bereits bekannte Angriffe direkt zu identifizieren. Unter Umständen ist außerdem der *Policy Script Interpreter* um neue Sprachkonstrukte zu erweitern, um etwa den Zugriff auf übertragungsprotokollspezifische Informationen zu vereinfachen. Schließlich hat eine Interpretation der erzeugten policy-neutralen Ereignisse auf Grundlage des verwendeten Übertragungsprotokolls zu erfolgen.

## 4.3 Anpassung von Bro an Wireless LANs

Nach dieser allgemeinen Vorstellung von Bro wird nun seine Verwendung in mobilen Netzen beschrieben. Der entwickelte Prototyp dient der Erkennung von aktiven Angriffen auf ein Wireless LAN nach IEEE Standard 802.11. Die Wahl fiel auf diesen

Übertragungsstandard, da er sowohl einen Infrastruktur- als auch einen Ad-hoc-Modus beschreibt. Darüber hinaus erfreut er sich weiter Verbreitung, was auch zu einer Vielzahl frei verfügbarer Angriffswerkzeuge geführt hat, die im Rahmen der in Kapitel 4.4.5 beschriebenen Validierung genutzt werden.

Das Gros der Angriffe in infrastrukturellen Wireless LANs zielt auf die Verbindungsschicht des OSI-Referenzmodells, so dass sich auch die im Folgenden beschriebenen Erweiterungen hierauf konzentrieren. Im wesentlichen muss die Erfassung und Analyse von Management- und Control-Frames durch Bro ermöglicht werden, insbesondere der vollständige Zugriff auf die mitgeschnittenen Datenpakete einschließlich aller Informationen des Frame-Headers.

Die von Bro verwendete C-Bibliothek libpcap zum Erfassen des Netzverkehrs unterstützt bereits seit einiger Zeit den Empfang von 802.11-Frames und muss daher nicht verändert werden. Da den 802.11-Paketen allerdings ein eigener Datalink-Typ zugeordnet wird, müssen entsprechende Anpassungen an der Bro-Klasse PktSrc vorgenommen werden, in der die Schnittstelle zwischen Bro und der darunterliegenden Packet Capturing Bibliothek implementiert ist. Desweiteren muss die Bro-Komponente *Network Analysis* um die notwendige Funktionalität für die Verarbeitung von 802.11-Frames ergänzt werden.

Im Sinne der von Bro propagierten Verwendung einer domänenspezifischen Nomenklatur [Pax99, Kapitel 3] wird ferner die *Event Engine* um spezielle Ereignisse erweitert, die beim Einsatz von IEEE 802.11 auftreten können. Durch die Definition entsprechender *Event Handler* werden diese auf die *Policy-Ebene* abgebildet und so ihre Interpretation ermöglicht. Dies zieht natürlich auch den Ausbau der Bro Skriptsprache und des *Policy Script Interpreters* nach sich.

### 4.3.1 Erweiterung der Netzwerkanalyse und Ereignisgenerierung

Der Ausbau der Bro Netzwerkanalyse und Ereignisgenerierung für IEEE 802.11 lässt sich in die drei Teile Erfassung und Analyse von Audit-Daten sowie Definition WLAN-spezifischer Ereignisse untergliedern, die im Folgenden genauer beschrieben werden.

#### Erfassung der Audit-Daten

Die Klasse PktSrc repräsentiert die Schnittstelle zwischen Bro und der Bibliothek libpcap (vgl. Abbildung 4.3 auf Seite 76). Während der Initialisierung ermittelt sie den zugrunde liegenden Netzwerktyp der überwachten Netzschnittstelle, um so die Größe des Schicht 2 Protokoll-Headers zu bestimmen und diesen für die weitere Analyse zu verwerfen. Wie bereits erwähnt, enthält im Falle von IEEE 802.11 aber gerade dieser Header essentielle Informationen zur Erkennung von Angriffen und muss daher erhalten bleiben. Die Methode PktSrc::SetHdrSize wird daher so angepasst, dass für den Netzwerktyp 802.11 die Header-Größe auf Null gesetzt wird und somit der Layer 2 Header

erhalten bleibt. Um im weiteren Verlauf zwischen herkömmlichen und 802.11-Paketen unterscheiden zu können, wird außerdem eine Methode PktSrc::IsWireless ergänzt.

Bei der Analyse von 802.11-Verkehr ist außerdem der Übertragungskanal von Interesse, auf den die Netzwerkschnittstelle eingestellt ist. Daher wurde auch der Konstruktor der Klasse PktSrc erweitert, um den aktuellen Kanal der drahtlosen Schnittstelle zu ermitteln. Zum Zugriff auf diese Information wurde schließlich die Methode PktSrc::GetChannel ergänzt.

### Analyse der Audit-Daten

Die eigentliche Analyse des Netzverkehrs erfolgt üblicherweise durch die Klasse NetSessions, die jedoch nur die Schichten 3 bis 7 untersucht. Zur Verarbeitung der 802.11-Frames wird daher das ursprüngliche Bro um eine neue Klasse WLANSessions ergänzt, die entsprechende Pakete in ihre atomaren Bestandteile zerlegt und somit einen alternativen *Protocol Analyzer* darstellt. Die prinzipielle Vorgehensweise hierbei ist wie folgt und orientiert sich an dem in Abbildung 4.4 dargestellten Aufbau von 802.11 MAC-Frames:

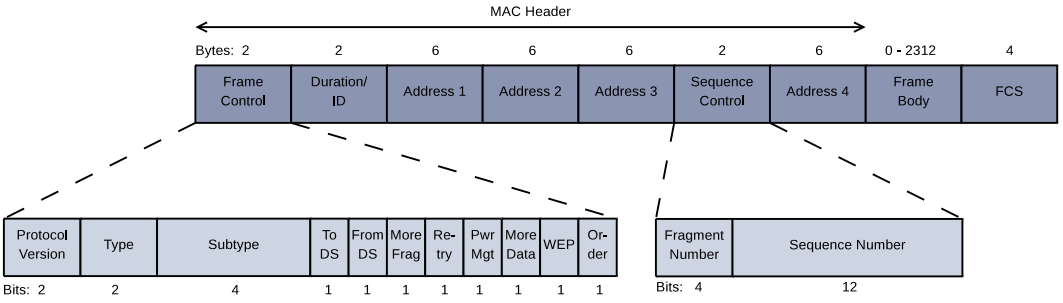



Abbildung 4.4: Das Format von 802.11 MAC Frames gemäß [LAN99]

1. Zunächst wird der konkrete Typ des aktuell untersuchten 802.11-Pakets bestimmt. Der IEEE Standard 802.11 unterscheidet dabei grundsätzlich drei Nachrichtentypen:

   *Kontroll-Nachrichten* (engl. *Control Frames*): Hierbei handelt es sich um kurze Nachrichten zur Koordination des Sendekanals und zur Signalisierung von Kommunikationsfehlern.

   *Management-Nachrichten* (engl. *Management Frames*): Mit Nachrichten dieses Typs können die einzelnen Netzknoten ihre Kommunikationsbeziehungen aushandeln und kontrollieren. Hierzu gehören beispielsweise die Nachrichten, mit denen ein mobiles Endgerät Zugang zu einem Access Point erbittet.

*Daten-Nachrichten* (engl. *Data Frames*): Nachdem die Knoten solche Kommunikationsbeziehungen etabliert haben, können sie Nutzdaten mittels dieses Nachrichtentyps austauschen.

Für eine umfassende Darstellung der möglichen Ausprägungen der einzelnen Frame-Typen und -Subtypen sei auf [LAN99, Tabelle 1 auf Seite 36] verwiesen.

2. Anschließend wird der MAC-Header in seine Bestandteile zerlegt, um etwa die enthaltenen Adressinformationen zu bestimmen.

3. Vor der weiteren Untersuchung des eigentlichen Datenteils (Frame Body) werden fragmentierte Pakete zunächst wieder zusammengesetzt.

4. Neben den Daten des Frame-Headers sind auch die Informationen des Frame-Body von Management-Frames von Interesse. Bei ihrer Extraktion muss zwischen obligatorischen Feldern mit fester Länge und optionalen Feldern mit variabler Länge unterschieden werden.

5. Nach der nun erfolgten Zerlegung des Pakets erfolgt seine Analyse und die Generierung entsprechender Ereignisse.

Die Zerlegung und Untersuchung der 802.11-Frames orientiert sich dabei ganz bewusst am ursprünglichen Standard IEEE 802.11 und lässt Erweiterungen wie etwa IEEE 802.11a, b, g oder n außer acht, da das von ihnen verwendete Frame-Format kompatibel zu IEEE 802.11 ist und auch gängige Angriffe nicht auf die Besonderheiten der Substandards eingehen. Eine entsprechende Erweiterung ist aber problemlos möglich. Gleiches gilt für weitergehende Sicherheitsstandards, wie etwa WPA oder IEEE 802.11i.

Der neue *Protocol Analyzer* wird wie in Abbildung 4.5 auf der nächsten Seite dargestellt in das bestehende Bro Framework integriert. Neu hinzugefügte bzw. modifizierte Komponenten sind hervorgehoben. Mit Hilfe der bereits erwähnten Methode PktSrc::IsWireless wird in der Methode PktSrc::ExtractNextPacket ermittelt, ob das betrachtete Paket über eine drahtlose Netzwerkverbindung empfangen wurde. Falls ja, übernimmt die Klasse WLANSessions die weitere Analyse, andernfalls wird die herkömmliche Verarbeitung durch die Klasse NetSessions gewählt.

#### Definition WLAN-spezifischer Ereignisse

Die Definition spezifischer Ereignisse für WLANs ist keine einfache Aufgabe, da sich im Vorfeld nur schwer abschätzen lässt, welche konkreten Ereignisse zur Erkennung bestimmter Angriffe notwendig sind. Auch fordert der Bro-Ansatz die strikte Neutralität der Ereignisse, d. h. auch die implizite Deutung eines Ereignisses ist zu diesem Zeitpunkt unbedingt zu vermeiden. Es werden daher zunächst Ereignisse zur Unterscheidung der Basistypen der empfangenen 802.11-Frames festgelegt. Zusammen mit einem Fehlerereignis für nicht standardkonforme Frames ergeben sich somit die vier folgenden

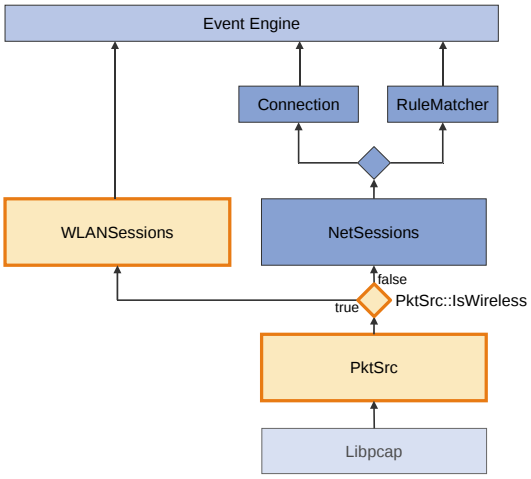



Abbildung 4.5: Erweiterung der Bro Network Analysis für IEEE 802.11 Pakete

Ereignisse, wobei der jeweilige Frame-Subtyp und der bei der Übermittlung benutzte Kanal ebenfalls übergeben werden:

- wlan_data_frame(subtype:count, channel:count)
- wlan_control_frame(subtype:count, channel:count)
- wlan_management_frame(subtype:count, channel:count)
- wlan_unknown_type(type:count, subtype:count, channel:count)

Eine Betrachtung bekannter Angriffe auf Wireless LANs zeigt darüber hinaus, dass in den meisten Fällen gefälschte Management-Frames verwendet werden (vgl. hierzu auch [VGM04] und Abbildung 2.5 auf Seite 22). Dies ist wenig verwunderlich, da die Koordination der Sicherheitsmechanismen ausschließlich auf Management-Ebene abläuft [EA04b]. Im folgenden werden daher weitere Ereignisse zur Differenzierung einzelner Management-Nachrichten festgelegt, um eine genauere Analyse der betreffenden Protokollabläufe und somit eine bessere Erkennung darauf zielender Angriffe zu ermöglichen. Tabelle 4.2 auf der nächsten Seite fasst die generierten Management-Ereignisse nebst der jeweils aus dem Frame Body ermittelten und mit ihnen an die *Event Engine* übergebenen Parameter zusammen. Deren genaue Bedeutung wird beispielsweise in [Rec04] näher erläutert.

Darüber hinaus wird wie schon bei den vier Basisereignissen auch der benutzte Übertragungskanal und außerdem der Header des Management Frames übergeben. Die im Management Frame Header enthaltenen Informationen sind in Tabelle 4.3 auf Seite 83 zusammengefasst.

| Ereignis | Übergebene Parameter | Erläuterung |
|---|---|---|
| wlan_assoc_req | SSID | An- und Abmeldung beim Access Point |
| wlan_assoc_resp | Capability Information, Status Code, Association ID | |
| wlan_reassoc_req | SSID, Current AP Address | |
| wlan_reassoc_resp | Capability Information, Status Code, Association ID | |
| wlan_probe_req | SSID | Auffinden von Access Points |
| wlan_probe_resp | Beacon Intervall, Capability Information, SSID | |
| wlan_beacon | Beacon Intervall, Capability Information, SSID | |
| wlan_auth | Authentication Algorithm Number, Authentication Transaction Sequence Number, Status Code | Authentifizierung von Stationen |
| wlan_disassoc | Reason Code | |
| wlan_deauth | Reason Code | |

Tabelle 4.2: Differenzierende Ereignisse für IEEE 802.11 Management Frames

Das beschriebene Vorgehen zur Analyse des empfangenen Netzverkehrs betrachtet jedes einzelne Paket für sich. Ein alternativer Ansatz zur Generierung neuer Ereignisse ist die Realisierung eines erweiterten *Protocol Analyzers*, der die einzelnen 802.11 Pakete einer bestehenden Verbindung zuordnet und diese in Form eines Zustandsautomaten verwaltet. Jede Zustandsänderung der Verbindung würde dann zu einem entsprechenden Ereignis führen. Hierfür müssten jedoch am bestehenden Bro-System Änderungen in wesentlich größerem Umfang vorgenommen werden, als dies bei dem gewählten Ansatz der Fall war. Zudem lässt sich die Aggregation einzelner Pakete zu Verbindungen auch auf *Policy-Ebene* durchführen, wenn auch mit höherem Aufwand.

### 4.3.2 Erweiterung des Policy Script Interpreters

Alle Bestandteile der IEEE 802.11 Frame Header lassen sich grundsätzlich mit Hilfe der elementaren Datentypen beschreiben, die Bro bereits mitbringt. Es ist jedoch recht mühsam, die verwendeten MAC-Adressen als Textstring zu verwalten. Daher wurde die Bro Skriptsprache und damit auch der *Policy Script Interpreter* um den elementaren Datentyp mac zur intuitiven Handhabung eben dieser MAC-Adressen in kanonischer Darstellung erweitert.

Grundsätzlich zerlegt der Bro *Policy Script Interpreter* ein Policy-Skript in seine Bestandteile und bildet diese auf eine *Abstract Syntax Tree (AST)* genannte logische Baumstruktur korrespondierender C++ Objekte des Bro Kerns ab. Diese Struktur wird dann Stück für Stück ausgewertet, indem ausgehend von der Wurzel eines gegebenen Teilbaums eine

| Eintrag | Erläuterung |
|---|---|
| Frame Control | Festlegung des exakten Frame-Typs und Übermittlung von Kontrollinformationen (vgl. Abbildung 4.4 auf Seite 79) |
| Duration/ID | benötigte Dauer der Belegung des Übertragungsmediums zur Bestätigung des Frames |
| DA | Zieladresse des Frames (engl. *Destination Address*) |
| SA | Quelladresse des Frames (engl. *Source Address*) |
| BSSID | *Basic Service Set Identification* zur Unterscheidung verschiedener Funkzellen |
| Sequence Control | bestehend aus Fragment- und Sequenznummer zur Koordination des Nachrichtenstroms |

Tabelle 4.3: Aufbau des IEEE 802.11 Management Frames Headers

virtuelle Methode des entsprechenden Objekts aufgerufen wird. Die Erweiterung des *Policy Script Interpreters* um den neuen Datentyp mac gliedert sich damit in vier Schritte:

1. Die elementaren Typen der Bro Skriptsprache werden im Bro Kern durch eine Instanz der allgemeinen Klasse BroType repräsentiert. Die Zuordnung zu einem konkreten Typ erfolgt dabei über eine eindeutige Markierung, die durch den Datentyp TypeTag definiert wird. Um auf den neuen Datentyp referenzieren zu können, wird dieser um eine neue Markierung TYPE_MAC erweitert.

2. Die Festlegung des Wertebereichs eines elementaren Typs ist in der Klasse BroValUnion gekapselt, die ebenfalls um eine Behandlung des neuen Datentyps erweitert wird. Der Zugriff auf einen konkreten Wert erfolgt dabei über die Klasse Val, von der eine spezielle Klasse MacVal zur intuitiven Handhabung von MAC-Adressen in der gebräuchlichen hexadezimalen Notation abgeleitet wird.

3. Natürlich müssen auch sämtliche vorhandenen Basisoperatoren entsprechend erweitert werden, um Ausdrücke mit Variablen des neuen Typs auflösen zu können. Gleiches gilt für diverse globale Funktionen, etwa zur Überprüfung des Typs einer gegebenen Variablen.

4. Für die Verwendung von MAC-Adressen als Index von Bro-Tabellen ist schließlich noch die Klasse CompositeHash zu erweitern. Intern verwaltet Bro Tabellen als Hash Tables. Die Klasse CompositeHash stellt mit den Methoden ComputeHash, ComputeSingletonHash, RecoverVals und RecoverOneVal dafür die Funktionalität zur Berechnung eines Hash-Wertes für einen gegebenen Index-Wert und umgekehrt bereit. Sie sind für den neuen Datentyp entsprechend anzupassen.

### 4.3.3 Erweiterung der Policy-Ebene

Zur Verarbeitung der generierten Ereignisse müssen auf *Policy-Ebene* nicht nur entsprechende *Event Handler* festgelegt, sondern auch die von ihnen verwendeten globalen Variablen deklariert werden. Hierfür sind jedoch zunächst die neu hinzugefügten Datenstrukturen und -typen zu deklarieren. Listing 4.1 zeigt dies exemplarisch für Control und Management Frames. Die einzelnen Elemente der beiden Typen leiten sich direkt aus [LAN99] ab (vgl. Abbildung 4.4 auf Seite 79).

```
type ieee80211_frame_control: record {
    proto: count;
    typ: count;
    subtype: count;
    toDS: count;
    fromDS: count;
    more_frags: count;
    retry: count;
    pwr_mgmt: count;
    more_data: count;
    wep: count;
    order: count;
};

type ieee80211_mgmt_frame_hdr: record {
    fc: ieee80211_frame_control;
    duration: count;
    da: mac;
    sa: mac;
    bssid: mac;
    fragN: count;
    seqN: count;
};
```

Listing 4.1: Deklaration der Datenstrukturen für Control und Management Frames

Listing 4.2 auf der nächsten Seite zeigt beispielhaft einen *Event Handler* für die Verarbeitung eines durch eine *Association Response* ausgelösten Ereignisses. Ein Access Point sendet eine *Association Response* an einen anfragenden Client, wenn er dessen Verbindungswunsch stattgibt. Als Aufrufparameter erhält der *Event Handler* das entsprechende Paket sowie die Nummer des Kanals, auf dem die Verbindung stattfindet. Er extrahiert dann die Adressen des Access Points und des betreffenden Clients aus dem Paket (Zeilen 11 und 12) und erzeugt aus diesen, zusammen mit der Nummer des Kanals, einen neuen Eintrag in einer globalen Liste von Verbindungen (Zeile 14).

```
type connection_t: record {
        ap: mac;
        sta: mac;
        channel: count;
};

global connections: table[mac] of connection_t;

event wlan_assoc_response(c: ieee80211_assoc_response, channel: count) {
  local conn: connection_t;
  conn$ap = c$hdr$sa;
  conn$sta = c$hdr$da;
  conn$channel = channel;
  add connections[conn$sta] = conn;
  print fmt("Station %m connected to AP %m.", conn$sta, conn$ap);
}
```

Listing 4.2: Verarbeitung eines Association Response Event

Darüber hinaus können auf *Policy-Ebene* durch Deklaration weiterer *Event Handler* höherwertige Ereignisse definiert werden. Ein Beispiel hierfür zeigt Listing 4.3 auf der nächsten Seite. Die Definition des Ereignisses wlan_new_access_point erfolgt ab Zeile 17 und wird erst im weiteren Verlauf genauer erläutert. Wird der Empfang eines Beacon Frames registriert, überprüft der *Event Handler* wlan_beacon zunächst, ob dieser von einem ihm bereits bekannten Access Point stammt. Ist dies nicht der Fall, löst er das oben genannte Ereignis aus (Zeile 28).

## 4.4 Erkennung von Angriffen auf Wireless LANs

Die im vorangegangenen Abschnitt beschriebenen Erweiterungen ermöglichen lediglich die Nutzung des Bro Frameworks zur grundlegenden Analyse des Netzverkehrs in Wireless LANs. Hierzu werden die empfangenen Datenpakete zerlegt, kategorisiert und auf Grundlage des IEEE Standards 802.11 zu primitiven Ereignissen verarbeitet. Um jedoch auch tatsächlich Angriffe zu erkennen, also IDS-Funktionalität zu implementieren, bedarf es einer Interpretation der aufgetretenen Ereignisse. Mit anderen Worten, es müssen spezielle Richtlinien entwickelt werden, um unbedenklichen Netzverkehr von potentiell gefährlichem zu unterscheiden. Diese Richtlinien können in der Bro Skriptsprache beschrieben werden, wobei das Bro Framework sowohl anomalie- als auch missbrauchserkennende Herangehensweisen unterstützt. In der vorliegenden Arbeit wurde letztere gewählt, da es eine Vielzahl bekannter Angriffe und Angriffswerkzeuge

```
type ieee80211_beacon: record {
        hdr: ieee80211_mgmt_frame_hdr;
        beacon_interval: count;
        capabilities : count;
        ssid: string;
        channel: count;
};

type ap_t: record {
        ssid: ssid_t;
        sa: mac;
        channel: count;
};

global known_ap: set [ap_t];

event wlan_new_access_point(ap: ap_t) {
        ...
}

event wlan_beacon(c: ieee80211_beacon, channel: count) {
        local ap: ap_t;

        ap$ssid = c$ssid;
        ap$sa = c$hdr$sa;
        ap$channel = c$channel;

        if (ap !in known_ap_set) event wlan_new_access_point(ap);
        add known_ap [ap];
}
```

Listing 4.3: Definition von Ereignissen auf Policy-Ebene

für IEEE 802.11 gibt. Deren Funktionsweise und Verhalten wurde analysiert, um Muster des jeweiligen Angriffs zu identifizieren.

Im folgenden wird zunächst die Auswahl der betrachteten Angriffsszenarien begründet. Sie stützt sich sowohl auf eine Literatur- und Web-Recherche als auch auf die Analyse bekannter Angriffswerkzeuge. Es folgt eine genauere Beschreibung der einzelnen Szenarien, um mögliche Vorgehensweisen zu ihrer Erkennung zu gewinnen. Diese allgemeinen Strategien werden schließlich in neue Bro Policies für Wireless LANs überführt.

### 4.4.1 Betrachtete Angriffsszenarien

Von den ersten in [Wal00, BGW00, BGW01] geäußerten Zweifeln an der Sicherheit von WEP über die Schilderung eines prinzipiellen Angriffs auf den ihm zugrunde liegenden RC4-Algorithmus [FMS01] bis zu dessen praktischer Umsetzung [SIR01] verging nicht einmal ein Jahr. Seitdem wurden nicht nur die Angriffe auf WEP so weit verfeinert, dass ein solchermaßen gesichertes WLAN binnen weniger Minuten kompromittiert werden kann [BHL06, TWP07], sondern es gibt mittlerweile auch eine Vielzahl frei verfügbarer Angriffswerkzeuge, die dies selbst Laien ermöglichen [WVE]. Aktuelle Werkzeuge beschränken sich dabei nicht nur auf Angriffe auf den geheimen WEP-Schlüssel, sondern ermöglichen auch kompliziertere Szenarien wie etwa Denial-of-Service- und Man-in-the-Middle-Angriffe [BS03].

Tabelle 4.4 auf der nächsten Seite fasst die heute gebräuchlichsten Angriffswerkzeuge und die von ihnen verfolgten Angriffszielen zusammen. Sie reichen von der Analyse eines WLANs, also der Suche nach Netzknoten und dem Abhören (engl. *Sniffing*) des Übertragungsmediums, über Denial-of-Service- und Man-in-the-Middle-Angriffe bis hin zur Rekonstruktion des Schlüsselbitstroms oder dem vollständigen Brechen des geheimen WEP-Schlüssels. Unter den Angriffszielen wird jeweils auf das entsprechende Teilziel des in Abbildung 2.5 auf Seite 22 dargestellten allgemeinen Angriffsbaums für Wireless LANs verwiesen. Die Spalte *„aktiv“* gibt an, ob das Werkzeug dabei den Netzverkehr lediglich beobachtet oder auch manipuliert, indem es etwa gefälschte Pakete injiziert. Eine Erprobung der Werkzeuge und detaillierte Analyse ihrer Funktionsweise wurde in der Diplomarbeit [Leh07] vorgenommen.

Es ist augenscheinlich, dass heute für praktisch alle in Abbildung 2.5 auf Seite 22 genannten übergeordneten Angriffsziele frei verfügbare Exploits existieren. Zum Teil verfügen diese sogar über einfach zu bedienende graphische Nutzerschnittstellen, so dass sie problemlos auch von Laien genutzt werden können. Aber auch die Verwendung der verbleibenden Werkzeuge stellt einen durchschnittlich versierten Angreifer vor keine größeren Probleme. In der Regel genügen hierzu grundlegende Kenntnisse in der Bedienung von Unix-Systemen [Leh07]. Die potentielle Gefährdung durch diese Werkzeuge muss somit insgesamt als hoch eingestuft werden.

Ferner fällt auf, dass die überwiegende Mehrzahl der bekannten Exploits aktive Strategien anwendet und somit grundsätzlich für ein Intrusion Detection System zu erkennen sind. Dies hat im wesentlichen zwei Gründe:

Entweder setzt eine Angriffsstrategie das direkte Eingreifen des Angreifers zwingend voraus. So verwenden gängige DoS- und MitM-Angriffe gefälschte Management-Frames zum Erreichen des jeweiligen Angriffsziels (vgl. Abbildung 2.5 auf Seite 22).

Oder aber die Erfolgsaussichten eines Angriffs lassen sich durch die Beeinflussung des Netzverkehrs signifikant verbessern. Beispielsweise benötigten die ersten Angriffe zum Brechen des WEP-Schlüssels noch das Mitschneiden von etwa 4–6 Mio. Frames

| Werkzeug | Angriffsziel | | | | | aktiv |
|---|---|---|---|---|---|---|
| | DoS *1.4* | Sniffing *2.1* | MitM *2.2* | Key *2.3* | Stream *2.4* | |
| Kismet [Kis] | × | ✓ | × | × | × | × |
| Airsnort [Aird] | × | ✓ | × | ✓ | × | × |
| NetStumbler [Net] | × | ✓ | × | × | × | ✓ |
| airodump-ng [Aire] | × | ✓ | × | × | × | × |
| aircrack-ng [Aire] | × | × | × | ✓ | × | × |
| aireplay-ng [Aire] | ✓ | × | × | ✓ | × | ✓ |
| packetforge-ng [Aire] | ✓ | × | × | ✓ | × | ✓ |
| wesside-ng [Aire] | × | ✓ | × | ✓ | × | ✓ |
| Airsnarf [Airc] | × | × | ✓ | × | × | ✓ |
| AirPWN [Airb] | × | × | ✓ | × | × | ✓ |
| Airjack [Aira] | ✓ | × | ✓ | × | × | ✓ |
| FakeAP [Fak] | ✓ | × | × | × | × | ✓ |
| void11 [voi] | ✓ | × | × | × | × | ✓ |
| chopchop [Kor04a] | × | × | × | × | ✓ | ✓ |
| wesside [BHL06] | × | ✓ | × | ✓ | ✓ | ✓ |

✓ = erfüllt; × = nicht erfüllt

Tabelle 4.4: Angriffswerkzeuge für Wireless LANs

[FMS01, SIR04]. Moderne Angriffe konnten diese Zahl durch Inijzieren gefälschter ARP[4] Nachrichten auf weniger als 100.000 Frames reduzieren [TWP07].

Im folgenden werden vier mögliche Angriffsszenarien betrachtet. Sie decken die in Abbildung 2.5 auf Seite 22 dargestellten grundlegenden Angriffsziele ab, wobei einige Teilziele nur exemplarisch beleuchtet werden. Insbesondere werden ausschließlich aktive Angriffe untersucht. Zwar existieren in der Literatur Ansätze zur Erkennung passiver Angreifer in WLANs [SLO04], diese setzen jedoch die Verwendung standardkonformer Hardware durch den Angreifer voraus, die ein *Request-to-Send-Frame* stets mit einem *Clear-to-Send-Frame* beantwortet. Trifft dies nicht zu, d. h. verhält der Angreifer sich in jedem Fall vollkommen passiv, ist er grundsätzlich nicht zu erkennen [Pfi00].

[4] Address Resolution Protocol [Plu82]

### Störung der Funktionsfähigkeit des WLANs

Es gibt vielfältige Möglichkeiten zur Beeinträchtigung der Funktionsfähigkeit eines WLANs mittels sogenannter Denial-of-Service-Angriffe. Ziel dabei ist, die Verfügbarkeit eines WLANs zu stören und autorisierte Nutzer am Zugriff auf das drahtlose Netz zu hindern. Wie in Abbildung 2.5 auf Seite 22 dargestellt gibt es unterschiedliche Ansatzpunkte in der Protokollhierarchie, um solche Angriffe umzusetzen. Im folgenden werden beispielhaft DoS-Angriffe auf der Managementebene von IEEE 802.11 betrachtet.

Am weitesten verbreitet sind hierbei sogenannte Flooding-Angriffe, bei denen ein Netzknoten durch unsinnige Anfragen derart beschäftigt wird, dass er nicht mehr in der Lage ist reguläre Datenpakete zu verarbeiten. Dies geschieht in der Regel durch massenhaftes Versenden gefälschter Management-Nachrichten. Die am weitesten verbreiteten Ausprägungen dieses Angriffs werden im Folgenden näher erläutert:

- Beim *Deauthentication Flooding* sollen Clients daran gehindert werden, sich mit einem Access Point zu assoziieren. Dazu werden vom Angreifer im Namen des Access Points permanent Deauthentication-Nachrichten an den Client gesendet, um diesen zu zwingen, sich erneut mit dem Access Point zu verbinden. Durch die ständig neu eintreffenden Deauthentication-Nachrichten wird jedoch eine stabile Verbindung unmöglich gemacht. Denselben Effekt erzielt der Angreifer auch durch das Versenden von Disassociate-Nachrichten.

- Ähnlich funktioniert auch das *Authentication Flooding*, wobei hier nicht der Client sondern der Access Point das Angriffsziel ist. Dazu werden massenhaft Authentication-Anfragen mit zufällig gewählten Absenderadressen an einen Access Point gesendet, um dessen Systemressourcen derart zu erschöpfen, dass er nicht mehr in der Lage ist Verbindungsanfragen regulärer Clients zu verarbeiten. Auch hier lässt sich ein vergleichbarer Effekt durch das Versenden gefälschter Association Requests erzielen.

- Das *Beacon Flooding* ist schließlich als Angriff auf das Netzwerk selbst zu werten. Hierbei versendet ein Angreifer massenhaft Beacon-Frames mit wechselnder SSID und Absenderadresse, um die verfügbare Übertragungskapazität für die anderen Stationen eines WLANs zu reduzieren. Zudem wird durch diese „Verschmutzung“ legitimen Clients die Auswahl eines geeigneten Access Points erschwert.

### Unzulässige und fehlerhaft konfigurierte (Rogue) Access Points

Unter Rogue Access Points werden unautorisiert in einem Firmennetz installierte oder fehlerhaft konfigurierte Access Points verstanden [Gei03]. Die Gründe für die unautorisierte Installation eines Access Points reichen dabei von der Bequemlichkeit einzelner Mitarbeiter bis hin zu gezielten Angriffen, etwa im Rahmen einer Man-in-the-Middle-Attacke. In jedem Fall stellen solche Geräte eine Gefährdung der Sicherheitsrichtlinien

eines lokalen Netzes dar, sei es, dass sie einen unkontrollierten Zugang zu einem gesicherten Netzbereich schaffen, oder aber direkt auf die Vertraulichkeit bzw. Integrität übermittelter Informationen zielen.

#### Zugriff auf Informationen mittels Man-in-the-Middle

Ein Man-in-the-Middle-Angriff zielt auf die Vertraulickeit und Integrität der Kommunikation zwischen Client und Access Point, indem deren Datenverkehr über den Angreifer umgeleitet wird. In Wireless LANs können dabei zwei grundsätzliche Varianten unterschieden werden [EA04a]:

Beim sogenannten *Hijacking* ersetzt ein Angreifer den Client *(Session Hijacking)* oder den Access Point *(Connection Hijacking)* einer bestehenden Verbindung zwischen einer Station und einem Access Point. Hierzu setzt er zunächst den jeweiligen Kommunikationspartner mittels eines DoS-Angriffs außer Gefecht bevor er dann seinen Platz einnimmt. Letzteres wird auch als *Maskerade* (engl. *Masquerading*) bezeichnet. Dazu sind die Identifikationsmerkmale des entsprechenden Kommunikationspartners (z.B. MAC-Adresse und SSID) zu imitieren.

Beim *vollständigen MitM* drängt sich der Angreifer in eine bestehende Verbindung (vgl. Abbildung 4.6 auf der nächsten Seite). Hierfür bringt er den Client etwa mittels gefälschter Deauthentication- oder Disassociation-Nachrichten zunächst dazu, die bestehende Verbindung mit dem regulären Access Point zu unterbrechen (Schritt 2). Versucht dieser, sich erneut zu verbinden, reißt der Angreifer als Rogue AP diese Verbindung an sich (Schritt 4.1). Dafür imitiert er den regulären Access Point (Schritt 3). Gegenüber diesem gibt er sich dann als Client aus (Schritt 4.2), so dass schließlich die gesamte Kommunikation zwischen den beiden regulären Geräten über ihn abläuft.

#### Bestimmung des Schlüsselbitstroms oder WEP-Schlüssels

Zugriff auf übermittelte Informationen kann ein Angreifer aber auch durch Ausnutzen der Schwächen des WEP-Protokolls und des ihm zugrundeliegenden RC4-Algorithmus erlangen. Hierbei lassen sich zwei grundlegende Arten von Angriffen unterscheiden [BHL06]:

*Wiederverwendung des Schlüsselbitstroms:* Schafft man es, den von RC4 generierten Schlüsselbitstrom zu rekonstruieren, so können alle mit ihm kodierten Pakete problemlos entschlüsselt werden. Ferner lassen sich mit ihm beliebige Pakete in ein Netz einschleusen.

*Brechen des WEP-Schlüssels:* Darüber hinaus hat der RC4-Algorithmus eine grundlegende Schwäche. Ist ein Angreifer in der Lage, genügend Pakete aufzufangen, in deren Verschlüsselung ein sogenannter schwacher Initialisierungsvektor (IV) eingeflossen ist, kann er damit den geheimen WEP-Schlüssel brechen [FMS01]. Der

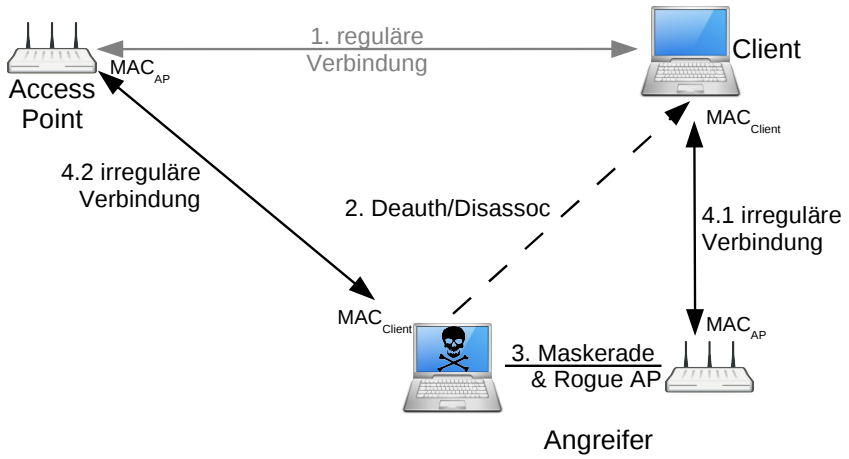



Abbildung 4.6: Prinzip eines vollständigen Man-in-the-Middle-Angriffs

kritische Faktor aus Sicht des Angreifers ist hierbei die Anzahl der benötigten Pakete, um ausreichend viele schwache Initialisierungsvektoren zu sammeln. Durch Anwendung statistischer Verfahren lässt sich diese reduzieren. So benötigt die in [Kor04b] beschriebene Variante lediglich Pakete mit eindeutigem IV.

Sowohl die Angriffe auf den Schlüsselbitstrom als auch auf den WEP-Schlüssel sind ursprünglich passiv angelegt, können aber wie bereits angesprochen durch Einschleusen gefälschter Nachrichten deutlich beschleunigt werden. Das hierfür notwendige Vorgehen wird beispielsweise in dem Exploit aireplay-ng [Aire] implementiert, dessen Funktionsweise im Folgenden kurz umrissen wird. Dabei muss zwischen zwei grundsätzlichen Ansätzen unterschieden werden.

Beim ersten fängt der Angreifer zunächst einen verschlüsselten ARP-Request ab. Ein solches Paket kann relativ problemlos an seiner geringen Größe und dem Umstand erkannt werden, dass es an die Broadcast-Adresse gerichtet ist. Da WEP-gesicherte WLANs über keinerlei Replay-Schutz verfügen (vergleiche Kapitel 2.4.1), kann nun dieser ARP-Request beliebig oft in das Netz eingespielt werden. Alle diese Anfragen werden mit einer ARP-Response beantwortet, wobei jede Antwort einen der benötigten eindeutigen IVs enthält. Dieses sogenannte *ARP Spoofing* verkürzt also die benötigte Zeit zum Sammeln eindeutiger Initialisierungsvektoren, so dass der Angriff des WEP-Schlüssels auch in Netzen mit geringer Auslastung oder bei wechselnden Schlüsseln möglich wird.

Der zweite Ansatz wurde unter dem Namen chopchop [Kor04a] bekannt und zielt auf die Rekonstruktion eines Schlüsselbitstroms samt IV. Mit diesen Informationen lassen sich ein beliebiges Datenpaket ohne Kenntnis des WEP-Schlüssels dechiffrieren oder aber

ein gültiges WEP-Paket zur Generierung weiteren künstlichen Verkehrs konstruieren. Der Angreifer schneidet hierzu das letzte Byte eines beliebigen WEP-Pakets ab. Unter der Annahme, dass dieses Byte 0 war, rekonstruiert er eine gültige Prüfsumme, wobei er sich die bereits in Kapitel 2.4.1 genannte Linearität des verwendeten CRC-Algorithmus zu Nutze macht. Das so konstruierte Paket sendet er an den Access Point und beobachtet, ob es von diesem akzeptiert wird. Das beschriebene Vorgehen wiederholt er für alle 256 möglichen Annahmen bzw. bis ein Paket akzeptiert wird. Um die injizierten Pakete voneinander unterscheiden zu können, wird die jeweils getroffene Annahme in der Ziel MAC-Adresse des Pakets kodiert. Der gesamte Vorgang wird dann für jedes weitere Byte des Pakets wiederholt, bis schließlich der komplette Schlüsselstrom rekonstruiert wurde.

### 4.4.2 Verwendete Methoden zur Angriffserkennung

Nachdem die in dieser Arbeit betrachteten Angriffsszenarien festgelegt sind, sollen nun allgemeine Strategien zu ihrer Erkennung vorgestellt werden. Ihre Anwendung in Bro Policies ist dann Gegenstand des folgenden Teilkapitels.

Amoroso nennt als grundlegende Funktion eines IDS die Korrelation, d. h. die „Interpretation, Kombination und Analyse aller verfügbaren Informationen über die Aktivität eines Systems zum Zwecke der Angriffserkennung und -erwiderung“ [Amo99b]. Im folgenden werden zunächst schwellwertbasierte Verfahren als das im wesentlichen verwendete Analyseinstrument eingeführt. Hinsichtlich der dabei herangezogenen Informationsquellen wird allgemein zwischen In-Band und Out-of-Band Informationen unterschieden [Amo99a, Bac00a].

#### Schwellwertbasierte Ansätze

Die im vorangegangenen Kapitel beschriebenen Angriffsszenarien basieren in vielen Fällen auf dem massenhaften Einschleusen gefälschter Management-Frames. Es ist daher naheliegend, bestimmte Eigenschaften des Netzverkehrs statistisch zu analysieren und außergewöhnliche Abweichungen als Anzeichen eines Angriffs zu werten. Mögliche Aspekte sind hierbei die durchschnittliche Anzahl gesendeter Frames eines bestimmten Typs oder aber auch die der in einem Netz verwendeten MAC-Adressen.

Das Hauptproblem bei solchen schwellwertbasierten Verfahren ist die möglichst optimale Festlegung, was als außergewöhnliche Abweichung vom durchschnittlichen Netzverkehr angesehen wird. Wird dieser Schwellwert zu niedrig gewählt, wertet das IDS unter Umständen zulässigen Verkehr als Angriffsversuch (*False Positive*). Wählt man ihn zu hoch, können Angriffe übersehen werden (*False Negative*). Ebenso schwierig ist die Wahl des betrachteten Zeitintervalls. Ist es zu lang, bleiben etwa kurzfristige Flooding-Angriffe unter Umständen unerkannt. Wählt man es zu kurz, steigt die Anzahl der False Positives, deren Bearbeitung unnötig Ressourcen beansprucht.

In der Praxis erfordert die Festlegung der Schwellwerte daher nicht nur ein tiefgreifendes Verständnis der betrachteten Netzprotokolle, sondern auch eine fundierte empirische Grundlage. Darüber hinaus kann die Genauigkeit schwellwertbasierter Verfahren durch Kombination mit anderen Analysemethoden verbessert werden. Ein Beispiel hierfür wird im folgenden Abschnitt gegeben.

#### In-Band Informationen

Unter In-Band Informationen werden dem betrachteten Zielsystem inhärente Audit-Daten verstanden [Amo99c]. Im Fall des hier verfolgten netzwerkbasierten Ansatzes handelt es sich dabei um die aus den übermittelten Netzwerkpaketen gewonnenen Informationen, insbesondere die aus dem Paket-Header.

Exemplarisch seien hierbei die auftretenden MAC-Adressen und Sequenznummern genannt. Beide lassen sich zur Erkennung eines MitM-Angriffs nutzen. So können anhand der MAC-Adressen die Verbindungskontexte einzelner Netzknoten hergeleitet werden, um scheinbare Mehrfachverbindungen eines Knotens mit einem Access Point zu identifizieren. Ferner können durch Analyse der mit einer bestimmten Absender-MAC-Adresse verwendeten Sequenznummern MAC-Spoofing-Angriffe aufgedeckt werden [Wri03, GC05].

#### Out-of-Band Informationen

Eine weitere wertvolle Informationsquelle zur Präzisierung der Angriffserkennung stellen aber auch sogenannte Out-of-Band Informationen dar. Diese Kategorie umfasst sämtliche außerhalb des eigentlichen IDS vorliegenden Informationen, auf die das IDS automatischen oder manuellen Zugriff hat [Bac00c].

Ein Beispiel der verwendeten Out-of-Band Informationen ist eine Liste gültiger MAC-Adressen. So beschreiben die ersten drei der insgesamt 6 Byte langen MAC-Adresse eine von der IEEE vergebene eindeutige Herstellerkennung (engl. *Organizationally Unique Identifier – OUI*). Von einem Angreifer frei erfundene MAC-Adressen können so unter Umständen erkannt werden.

Ein weiteres Beispiel stellen Listen zulässiger bzw. unzulässiger Access Points dar. Man spricht hierbei auch von *White* bzw. *Black Lists*. Diese lassen sich zum Teil automatisch aus vorhandenen Netzwerkmanagement- und -inventarisierungssystemen extrahieren und werden im laufenden Betrieb durch das IDS aktualisiert. Auf diese Weise können Rogue Access Points identifiziert werden. Sogenannte *Grey Lists* erfassen legitime Access Points in der Nachbarschaft des betrachteten WLANs, um die False-Positive-Rate zu reduzieren.

### 4.4.3 Entwicklung neuer Bro Policies für Wireless LANs

Die folgenden Bro Policies orientieren sich an den eingangs beschriebenen Angriffsszenarien. Sie belegen exemplarisch die Möglichkeiten des bereitgestellten Instrumentariums für die Erkennung von Angriffen in Wireless LANs. Natürlich können mit dem verfolgten missbrauchserkennenden Ansatz nur bekannte Angriffe erkannt werden. Insbesondere kann der Angreifer – so er denn Kenntnis von der Existenz des IDS hat – sein Verhalten anpassen, um die Richtlinien zu unterwandern. Dies ist jedoch ein grundsätzliches Problem von Intrusion Detection Maßnahmen. Kurz, es wird nicht der Anspruch erhoben, dass die entworfenen Richtlinien die Ultima Ratio darstellen. Nichtsdestoweniger wird die nachfolgende experimentelle Validierung zeigen, dass sie in der Tat ausreichen, um die Verwendung heute üblicher Exploits aufzudecken.

#### Denial-of-Service-Angriffe auf IEEE 802.11 Managementfunktionen

Stellvertretend für die Erkennung von Denial-of-Service-Angriffen unter Ausnutzung der Managementfunktionalität des IEEE 802.11 Standards wird im Folgenden *Authentication Flooding* untersucht. Wichtigstes Merkmal zur Erkennung eines solchen Angriffs ist ganz offensichtlich die ungewöhnlich hohe Anzahl in einem kurzen Zeitraum versendeter Management-Frames. Listing 4.4 auf der nächsten Seite zeigt die Umsetzung dieses Ansatzes in einem Bro Policy-Skript.

Über einen Zeitraum von 10 Sekunden (Zeile 1) wird die Anzahl der aufgetretenen Authentication Frames bestimmt. Dazu wird bei jedem Auftreten eines solchen Frames der Zähler auth_counter inkrementiert (Zeile 14). Überschreitet der Zähler am Ende des betrachteten Intervalls einen vorgegebenen Schwellwert, in unserem Beispiel 100 Stück (Zeile 3), so wird dies als Anzeichen für einen Flooding-Angriff gewertet und Alarm ausgelöst (Zeile 6 f.). Andernfalls wird der Zähler zurückgesetzt und von vorne begonnen.

Um automatisch am Ende des betrachteten Prüfintervalls die Häufigkeit der aufgetretenen Authentication Frames zu überprüfen wird der Befehl schedule verwendet, der den zeitgesteuerten Aufruf eines Ereignisses ermöglicht. Dies geschieht erstmalig während der allgemeinen Systeminitialisierung (Zeile 18) und erneut nach jeder erfolgten Überprüfung (Zeile 10).

Darüber hinaus kann die MAC-Adresse des Absenders eines solchen gefälschten Frames ein weiteres Indiz liefern, da sie in der Regel zufällig gewählt wurde und deshalb keinen gültigen OUI darstellt. Diese kann entweder händisch oder automatisch durch Verwendung externer Netzmanagement-Software aktualisiert werden, die via *Broccoli* an das Bro Framework angebunden wird.

```
global flooding_interval: interval = 10 sec;
global auth_counter: count = 0;
global auth_threshold: count = 100;

event wlan_check_AuthFlooding() {
        if ( auth_counter >= auth_threshold ) {
                print fmt("Alert: Threshold for authentication frames exceeded. ➥
                ➥ Possible flooding attack!") ;
        }
        auth_counter = 0;
        schedule flooding_interval { wlan_check_AuthFlooding() };
}

event wlan_auth(c: ieee80211_auth, channel: count) {
        auth_counter = auth_counter + 1;
}

event bro_init() {
        schedule flooding_interval { wlan_check_AuthFlooding() };
}
```

Listing 4.4: Erkennung eines Authentication Flooding Angriffs

### Aktive Angriffe des Schlüsselbitstroms und des WEP-Schlüssels

Die Erkennung aktiver Angriffe des Schlüsselbitstroms erfolgt ähnlich. Die in Listing 4.5 auf der nächsten Seite dargestellte Richtlinie zur Erkennung des chopchop Angriffs [Kor04a] basiert auf der durchschnittlichen Anzahl unterschiedlicher MAC-Adressen in einem Wireless LAN. Zur Erinnerung: In dieser wird die von chopchop getroffene Annahme über das rekonstruierte Byte eines abgefangenen Frames kodiert.

In den Zeilen 1 bis 10 wird zunächst ein neuer Datentyp zur allgemeinen Verarbeitung von WLAN-Frames definiert. Er basiert auf der Darstellung in Abbildung 4.4 auf Seite 79. Es folgen die globalen Variablen zur Bestimmung der Länge des Prüfintervalls, der Anzahl der in diesem Zeitraum beobachteten unterschiedlichen MAC-Adressen sowie der hierfür maximal zulässigen Zahl.

Die eigentliche Prüfroutine ist wie bereits in Listing 4.4 zweigeteilt. Für jedes aufgefangene WLAN-Frame wird überprüft, ob der Sender während des Prüfintervalls bereits registriert wurde (Zeile 22). Wenn nicht, wird dies nachgeholt und die Anzahl der aktiven Stationen erhöht. Die zur Verwaltung der beobachteten MAC-Adressen verwendete Liste wurde hierfür in Zeile 16 so deklariert, dass sie nach Ablauf des Prüfintervalls automatisch reinitialisiert wird.

Die Auswertung der ermittelten Daten erfolgt in den Zeilen 28–34, indem wiederum die während des Prüfzeitraums ermittelte Anzahl von MAC-Adressen mit dem zulässigen Höchstwert verglichen und gegebenenfalls Alarm ausgelöst wird.

```
1  type ieee80211_general_frame_hdr: record {
           fc: ieee80211_frame_control;
3          duration: count;
           addr1: mac;
5          addr2: mac;
           addr3: mac;
7          fragN: count;
           seqN: count;
9          addr4: mac;
   };
11
   global check_interval: interval = 1 sec;
13 global mac_threshold: count = 50;
   global mac_counter: count = 0;
15
   global mac_set: set[mac] & create_expire = check_interval;
17
   event wlan_general_frame(c: ieee80211_general_frame_hdr) {
19         local current_mac: mac;
           current_mac = c$addr2;
21
           if ( current_mac !in mac_set ) {
23                 add mac_set [current_mac];
                   mac_counter = mac_counter + 1;
25         }
   }
27
   event wlan_check_chopchop() {
29         if ( mac_counter >= mac_threshold ) {
                   print fmt("Alert: Threshold for new MAC addresses within given ➥
                   ➥ time frame exceeded. Possible chopchop attack!");
31         }
           mac_counter = 0;
33         schedule check_interval { wlan_check_chopchop(); };
   }
35
   event bro_init() {
37         schedule check_interval { wlan_check_chopchop(); };
   }
```

Listing 4.5: Erkennung eines chopchop Angriffs

Aktive Angriffe auf den WEP-Schlüssel basieren, wie bereits erwähnt, auf der Einspielung einer Vielzahl gefälschter ARP-Requests, um genügend eindeutige Initialisierungsvektoren zu gewinnen. Ein solcher Angriff lässt sich also wiederum durch die Verwendung eines entsprechenden Schwellwerts detektieren. Darüber hinaus geht diesem ARP Spoofing in der Praxis meist ein DoS-Angriff auf den Access Point voraus, um diesen zu einer Reinitialisierung und damit der Unterbrechung sämtlicher bestehender Verbindungen zu zwingen. Hierbei werden in der Regel auch die gespeicherten ARP-Tabellen gelöscht, so dass der Angreifer die beim anschließenden Neuaufbau der Verbindungen versandten ARP-Requests als Muster für seine gefälschten Nachrichten abfangen kann.

**Unzulässige Access Points (Rogue AP)**

Illegitime Access Points unterscheiden sich von den bisher beschriebenen Angriffsszenarien dahingehend, dass hinter ihnen nicht zwangsläufig ein egoistisch oder böswillig handelnder Angreifer stehen muss. Stattdessen kann auch „nur“ eine Fehlkonfiguration vorliegen. Dennoch stellen diese sogenannten Rogue Access Points eine nicht zu unterschätzende Gefahr dar. Zum einen ermöglichen sie einen vom Netzmanagement unberücksichtigten Zugang zu einem lokalen Netz, wobei möglicherweise zentrale Sicherheitseinrichtungen wie etwa Firewalls umgangen werden. Zum anderen werden sie als Baustein in komplexeren Angriffsszenarien wie etwa dem im folgenden Abschnitt beschriebenen Man-in-the-Middle-Angriff verwendet. Ihrer Erkennung wurde daher gerade auch in den kommerziellen Wireless IDS große Aufmerksamkeit gewidmet. Die hierzu im Folgenden vorgestellte Strategie orientiert sich an dem in [SLO04] vorgestellten Ansatz.

Die Unterscheidung zwischen legitimen und illegitimen Access Points muss letztlich durch den Nutzer erfolgen. Aus Sicht des IDS bedarf es hierfür also Out-of-Band Informationen. Diese werden in Listing 4.6 auf der nächsten Seite in drei Listen verwaltet:

- In Zeile 1 wird eine *Whitelist* aller legitimen Access Points definiert, d. h. jener Access Points, mit denen sich ein Client verbinden darf. Dies sind beispielsweise sämtliche Access Points eines Firmennetzes.
- Zeile 2 definiert darüber hinaus eine *Greylist* aller semi-zulässigen Access Points. Dabei handelt es sich um alle Access Points, mit denen ein mobiler Client zwar immer wieder in Kontakt kommt, zu denen er aber keine Verbindung aufbauen darf. Dies sind beispielsweise die Access Points angrenzender Wireless LANs, etwa in benachbarten Gebäuden. Die Verwendung dieser Greylist dient dabei ausschließlich der Reduktion von *False Positives*.
- Die in Zeile 3 definierte *Blacklist* sammelt alle erkannten Rogue APs und stellt sie für weitergehende Analysen zur Verfügung.

```
global trusted_ap: set [ap_t];
global neighbourhood_ap: set [ap_t];
global rogue_ap: set [mac, count];

event wlan_new_access_point(ap: ap_t) {
        if (ap !in trusted_ap && ap !in neighbourhood_ap) {
                add rogue_ap [ap$sa, ap$channel];
                print fmt("Alert: Rogue Access Point detected!");
        }
}
```

Listing 4.6: Erkennung eines Rogue AP

Der *Event Handler* wlan_new_access_point(ap: ap_t) erweitert die in Listing 4.3 auf Seite 86 vorgestellte Behandlung aufgefangener Beacon Frames. So wird bereits dort zwischen bekannten oder neuen Access Points unterschieden, um unnötige Meldungen zu vermeiden. Bei erstmalig in Erscheinung getretenen Access Points wird hier nun mit Hilfe der erwähnten White- und Greylist überprüft, ob es sich um einen legitimen AP handelt oder nicht. Wenn nicht, wird der neu erkannte Rogue AP der Blacklist hinzugefügt und eine Alarmmeldung generiert.

#### Man-in-the-Middle-Angriffe

Als letztes Szenario wird nun die Erkennung eines vollständigen Man-in-the-Middle-Angriffs beschrieben. Im Gegensatz zu den bisher betrachteten Angriffen kombiniert dieser verschiedene Basisangriffe. Wie bereits in Abbildung 4.6 auf Seite 91 dargestellt, führt ein MitM-Angreifer folgende Aktionen aus:

1. Unterbrechung der regulären Verbindung zwischen Client und Access Point durch das Versenden gefälschter Deauthentication- bzw. Disassociation-Nachrichten im Namen des legitimen Access Points.

2. Bereitstellung eines Rogue AP. Dessen Konfiguration ist quasi identisch zu der des legitimen Access Points. Sie unterscheidet sich lediglich in dem gewählten Übertragungskanal. Dies entspricht aber der üblichen Konfiguration einer *Extended Service Set (ESS)*.

3. Sobald sich der Client mit dem Rogue AP des Angreifers verbunden hat, assoziiert sich dieser wiederum mit dem regulären AP. Dabei imitiert er den legitimen Client, indem er dessen MAC-Adresse verwendet.

Alle drei genannten Aktionen greifen aktiv in den Netzverkehr ein und sind damit grundsätzlich für ein IDS zu erkennen. Im folgenden wird daher das „gleichzeitige“

Eintreten folgender Ereignisse als Indiz für eine gerade durchgeführte MitM-Attacke gewertet:

*Unterbrechung einer regulären Verbindung:* Die Identifizierung eines hierfür verwendeten Flooding-Angriffs wurde bereits in Listing 4.4 auf Seite 95 exemplarisch vorgestellt.

*Endeckung eines Rogue AP:* Das Auffinden eines solchen illegitimen Access Point wurde im vorangegangenen Abschnitt erörtert.

*Verbindung Client–Rogue AP:* Der Aufbau einer Verbindung zu einem Client wird von einem Access Point durch Aussenden eines Association Response Frames abgeschlossen. Auf dieser Grundlage lassen sich neue Verbindungen erkennen (vergleiche auch Listing 4.2 auf Seite 85). In Listing 4.7 wird nun dieses Prozedere um einen Vergleich mit der bereits zuvor in Listing 4.6 auf der vorherigen Seite eingeführten Tabelle bekannter Rogue Access Points erweitert.

```
event wlan_assoc_resp(c: ieee80211_assoc_resp, channel: count) {
    if ( [c$hdr$sa, channel] in rogue_ap ) {
        print fmt("Alert: Station %m connected to Rogue AP %m.", c$hdr$da,
            c$hdr$da);
    }
}
```

Listing 4.7: Erkennung des Verbindungsaufbaus zu einem Rogue AP

*Verbindung Angreifer–Access Point:* Um nun außerdem die zeitgleiche Verbindung des (scheinbar) selben Clients mit einem legitimen AP zu erkennen, muss darüber hinaus für die einzelnen Clients ein Verzeichnis ihrer bestehenden Verbindungen geführt werden. Die hierfür verwendete Datenstruktur connections wurde bereits in Listing 4.2 auf Seite 85 (Zeile 7) definiert, ebenso die Aktualisierung des Verzeichnisses nach Empfang eines Association Response Frames, das den Beginn einer Verbindung markiert. Listing 4.8 auf der nächsten Seite zeigt nun die Bereinigung des Verzeichnisses nach Beendigung einer Verbindung, was durch den Empfang eines Disassociation Frames angezeigt wird.

Unter der Annahme, dass der IDS-Monitor sämtliche Association Response und Disassociation Frames empfängt, kann somit der Verbindungsstatus jeder Station erfasst werden. Genau dies ist in der Praxis jedoch nicht immer gegeben, weshalb es zu Fehlinterpretationen kommen kann. Zur Verbesserung der Erkennungsleistung wird daher im Folgenden das bisher ausschließlich lokal operierende System zu einem verteilten ausgebaut.

```
event wlan_disassoc(c: ieee80211_disassoc, channel: count) {
        if ( c$hdr$sa == c$hdr$bssid) {
                print fmt("Association with station %m released by AP %m.", ➥
                ➥ c$hdr$sa, c$hdr$da);
                delete connections [c$hdr$da];
        }
        else {
                print fmt("Association with AP %m released by station %m.", ➥
                ➥ c$hdr$da, c$hdr$sa);
                delete connections [c$hdr$sa];
        }
}
```

Listing 4.8: Aktualisieren der Verbindungsübersicht nach Lösen einer Assoziation

### 4.4.4 Verteilte Erkennung in engen Kooperationsgruppen

Um die vorhandenen Ressourcen mobiler Endgeräte besser zu nutzen und ihre individuellen Sichten auf die sie umgebende Systemumgebung zu konsolidieren, wurde bereits in Kapitel 1.2 die Kooperation autonomer Endgeräte postuliert. In Kapitel 3.1.3 wurde der Gedanke zur Bildung solcher Kooperationsgruppen genannter Zusammenschlüsse mobiler Netzknoten weiter präzisiert. Die sich hieraus ergebenden Informationen können dabei je nach Standpunkt als In-Band oder Out-of-Band Informationen aufgefasst werden (siehe hierzu auch die entsprechenden Abschnitte in Kapitel 4.4.2). So stellt der Zusammenschluss einander vollkommen vertrauender mobiler Knoten zu einer engen Kooperationsgruppe die Bildung eines übergeordneten Systems dar. Gemäß der Definition in [Amo99b] sind damit alle in dieser Gruppe gemeinsam erarbeiteten und ausgetauschten Analyseergebnisse den In-Band Informationen zuzurechnen.

Lose Kooperationsgruppen betonen hingegen den individuellen Charakter jedes einzelnen beteiligten Systems, weshalb die ausgetauschten Informationen als Out-of-Band Informationen anzusehen sind. Die Herausforderung bei der Nutzung dieser Informationen liegt dabei in der Etablierung und Aufrechterhaltung von Vertrauensbeziehungen zwischen den einzelnen Netzknoten (vgl. hierzu auch [Amo99c, S. 31]).

Im folgenden wird zunächst die Realisierung enger Kooperationsgruppen beschrieben. Ihre Implementierung und Evaluierung erfolgte im Rahmen einer studentischen Belegarbeit [Rei06]. Die Gestaltung loser Kooperationen ist Gegenstand von Kapitel 5.

#### Austausch von Ereignissen

Mit der Komponente *Peer Communication I/O* (vgl. Abbildung 4.2 auf Seite 73) bietet das Bro Framework bereits eine grundlegende Unterstützung für die verteilte Erkennung von Angriffen. Die Komponente stellt im wesentlichen eine Implementierung

des Observer-Entwurfsmusters [GHJV95] dar und realisiert einen *Publish/Subscribe*-Mechanismus zur Verbreitung von Ereignisinformationen. Dazu abonniert ein *Beobachter* (engl. *Observer*) bei einer anderen IDS-Instanz, dem sogenannten *Subjekt*, alle für ihn relevanten Ereignisklassen. Wird in der Folge beim Subjekt ein entsprechendes Ereignis generiert, so benachrichtigt dieses alle registrierten Beobachter. Zur Absicherung der hierfür notwendigen Kommunikation wird optional SSL/TLS eingesetzt. Auf diese Weise lassen sich nicht nur Informationen zwischen verschiedenen mobilen Geräten austauschen, sondern auch weitergehende Analysen auf *Policy-Ebene* an andere Geräte delegieren. Die Konfiguration solcher engen Kooperationsgruppen erfolgt fest kodiert in den Policy-Skripten.

#### Verteilte Erkennung des Man-in-the-Middle-Angriffs

Der Nutzen einer solchen Zusammenarbeit soll exemplarisch an der verteilten Erkennung eines Man-in-the-Middle-Angriffs demonstriert werden. So setzt die im vorangegangenen Abschnitt vorgestellte Strategie zur Erkennung eines solchen Angriffs voraus, dass ein IDS-Monitor sämtliche Management Frames empfängt, um den korrekten Verbindungsstatus aller Stationen zu erfassen. Dies ist in der Praxis nicht immer zu gewährleisten. Zum einen können nur die Signale der Stationen innerhalb einer bestimmten Funkreichweite aufgefangen werden, zum anderen kann gängige WLAN-Hardware nur auf einem Sendekanal empfangen und ist deshalb nicht zur simultanen Überwachung des gesamten Frequenzspektrums in der Lage.

Stattdessen wird nun angenommen, dass alle Mitglieder einer engen Kooperationsgruppe sich gegenseitig als Observer registrieren und über einen gesicherten Kanal miteinander verbunden sind, über den ihre jeweiligen IDS-Instanzen Ereignisdaten austauschen.

Empfängt nun der Client aus Abbildung 4.6 auf Seite 91 den Deauthentication Frame des Angreifers (Schritt 2), so wird dies von seinem lokalen IDS registriert und an die bei ihm registrierten Observer weitergeleitet. Hierzu gehört auch das IDS des Access Points. Dieser vergleicht nun die Informationen der empfangenen Ereignismeldung mit seinem lokalen Datenbestand und kann damit feststellen, dass er das fragliche Deauthentication Frame nicht ausgesendet hat. Folgerichtig wird ein Angriff diagnostiziert und ein entsprechendes Ereignis generiert, welches wiederum an alle registrierten Observer und somit auch an das IDS des Clients weitergereicht wird. Das Vorgehen bei der Bestätigung eines Verbindungsaufbaus durch Versenden einer Association Response ist analog. Somit kann sowohl die Unterbrechung einer regulären Verbindung als auch die Verbindung zwischen einem regulären Client und einem Rogue Access Point bzw. zwischen einem Angreifer und dem regulären Access Point aufgedeckt werden. Dabei ist es unerheblich, ob die IDS-Instanz einer beteiligten Partei oder irgendeines anderen Mitglieds einer engen Kooperationsgruppe die fragliche Ereignismeldung generiert hat.

#### Implizite Adressierung

Ein wesentlicher Aspekt bei der Interpretation übermittelter Ereignismeldungen ist bis jetzt unerwähnt geblieben. Zwar werden alle Ereignisinformationen an sämtliche dafür registrierten Observer versendet, jedoch ist nicht unbedingt jeder Observer in der Lage, jede einzelne Information zu interpretieren. So kann im obigen Beispiel nur der reguläre Access Point entscheiden, ob er den fraglichen Management Frame versendet hat oder nicht. Um die für ihn interpretierbaren Ereignisse zu identifizieren greift daher jeder Observer auf die in den Header-Daten des ereignisauslösenden Management Frames enthaltenen Adressinformationen zurück, die als Teil des Ereignisses übermittelt werden.

#### Verbreitung von Angriffswarnungen

Ein weiterer Punkt ist das Propagieren von Angriffswarnungen über Kooperationsgruppengrenzen hinweg bzw. zwischen Endgeräten einer Kooperationsgruppe, die keine direkte Kooperationsbeziehung eingehen können. So kann ein Endgerät grundsätzlich Mitglied in mehreren Kooperationsgruppen sein. In Abbildung 3.2 auf Seite 64 sind beispielsweise alle Endgeräte von Alice zu einer Kooperationsgruppe zusammengefasst. Ihr PDA kooperiert darüber hinaus über eine Ad-hoc-Verbindung auch noch mit Dave und Bob.

Damit nun Alarmmeldungen von diesen auch Alices weitere Endgeräte erreichen, muss ihr PDA sie weiterleiten. Das gleiche gilt für Informationen von Bob innerhalb der zweiten Kooperationsgruppe, die von Dave an Alice weitervermittelt werden müssen. Zur Verbreitung von Angriffswarnungen wird ein Flooding-basierter Ansatz verwendet, um empfangene Alarmmeldungen an alle Bekannten eines Knotens weiterzuleiten. Diese Technik wird beispielsweise beim Broadcast Routing [KR07b] oder auch in Peer-to-Peer Netzen [SW05] verwendet. Für letztgenanntes Gebiet gibt es dabei auch eine Reihe von Untersuchungen, die diesem Ansatz eine schlechte Skalierbarkeit attestieren [And01, Rit01, Rip01]. Diese Problematik erscheint an dieser Stelle jedoch vernachlässigbar, da sich in engen Kooperationsgruppen vom Grundsatz her nur wenige Partner zusammenfinden.

### 4.4.5 Experimentelle Validierung

Ein Ziel der Arbeit war die Entwicklung der technischen Mittel zur Erkennung gängiger Angriffe in mobilen Netzen. Der vorgestellte Prototyp setzt dies für Wireless LANs um. Eine Bewertung des Prototypen kann grundsätzlich anhand dreier Parameter [Fle01, PZC+96] erfolgen:

*Genauigkeit* (engl. *Accuracy)*, gemessen in der Anzahl der False Positives.

*Performanz* (engl. *Performance)*, gemessen in der Verarbeitungsrate von Ereignissen.

*Vollständigkeit* (engl. *Completeness)*, gemessen in der Anzahl der False Negatives.

Mit wenigen Ausnahmen, die beispielsweise in [DM02] zusammengefasst werden, beschränkt sich die Untersuchung von IDS-Lösungen bisher meist auf singuläre Betrachtungen. Dabei versuchen alle bekannten Ansätze eine möglichst realistische Testumgebung nachzustellen, in der das betrachtete System überprüft wird.

Die im Folgenden beschriebene Validierung verfolgt denselben Ansatz, indem sie im Rahmen mehrerer Laborversuche die Vollständigkeit des vorgestellten Prototyps untersucht.[5] Um dabei möglichst praxisnahe Ergebnisse zu erhalten, wurde hierfür auf vorhandene Angriffswerkzeuge zurückgegriffen, die frei über das Internet verfügbar sind. Der konkrete Versuchsaufbau und die erzielten Ergebnisse werden im Folgenden beschrieben.

#### Versuchsaufbau

Die Validierung des Prototypen erfolgte getrennt für die lokale und die verteilte Erkennung von Angriffen. Für erstgenannte wurde die in Tabelle 4.5 beschriebene Testumgebung verwendet.

| System | Beschreibung |
|---|---|
| Angreifer | Debian GNU/Linux 3.1 (Kernel 2.4.29)<br>Netgear WG311 (Atheros Chipsatz, madwifi Treiber 15.05.2005)<br>Linksys WPC11 (Prism Chipsatz, hostap Treiber 0.4.1, airjack Treiber 0.6.6alpha)<br>ZCOM XI-325HP+ |
| IDS | Debian GNU/Linux 3.1 (Kernel 2.6.12)<br>Netgear WAG511 (Atheros Chipsatz, madwifi Treiber 15.05.2005)<br>Netgear MA521 (Realtek Chipsatz, Treiber von Andrea Merello version 0.21)<br>Netgear MA111 |
| Regulärer Client | Windows XP Professional (Service Pack 2)<br>Avaya Wireless Silver World Card (Hermes Chipsatz, Windows-Treiber) |
| Access Point | D-Link DI-624+<br>Netgear FWAG114 |

Tabelle 4.5: Konfiguration der Testumgebung

Sie besteht aus einem mobilen Client basierend auf einem handelsüblichen Windows-Laptop, der mit einem herkömmlichen Access Point verbunden ist. Access Point und mobiler Client wurden dabei räumlich getrennt platziert, um dem Angreifer einen Vorteil

[5] Aussagen zur Genauigkeit und Performanz werden zu diesem Zeitpunkt bewusst unterlassen. Sie erscheinen wenig aussagekräftig, wurde der Prototyp doch im Hinblick auf die gewählte Testumgebung optimiert.

hinsichtlich der Signalstärke zu verschaffen und so die Durchführung des Man-in-the-Middle-Angriffs zu vereinfachen. Sowohl Angreifer als auch IDS basieren auf einem Linux-System. Beide sind dabei mit zwei WLAN-Karten ausgerüstet. Der Angreifer, um im Falle des MitM-Angriffs simultane Verbindungen zum Client und zum regulären Access Point unterhalten zu können. Das IDS, um die beiden dabei genutzten Kanäle zu überwachen.

Für die Überprüfung der Funktionalität zur verteilten Erkennung wurde mit Hilfe der Virtualisierungssoftware VMware[6] eine Testumgebung bestehend aus einem Angreifer und drei bei der Angriffserkennung kooperierenden Systemen emuliert.

### Ergebnisse

Tabelle 4.6 fasst die durchgeführten Laborversuche und die dabei erzielten Ergebnisse zusammen. Die durchgeführten Angriffe decken die gesamte Bandbreite der in Kapitel 4.4.1 genannten Szenarien ab. Hierfür wurde eine Auswahl der in Tabelle 4.4 auf Seite 88 aufgelisteten Angriffswerkzeuge getroffen. Diese mussten entweder manuell auf dem Angreifersystem installiert werden oder aber sie ließen sich durch Verwendung der Linux Live-CD BackTrack Final 1.0 [7] direkt nutzen.

| Angriff | Angriffswerkzeug | erfolgreich | erkannt |
|---|---|---|---|
| Rogue Access Point | Airsnarf | ✓ | ✓ |
| Association-Flooding | void11 | × | ✓ |
| Authentication-Flooding | void11 | × | ✓ |
| Deauthentication-Flooding | Airjack | ✓ | ✓ |
| Beacon-Flooding | FakeAP | ✓ | ✓ |
| Man-in-the-Middle | Airjack | ✓ | ✓ |
| Schlüsselbitstrom | chopchop | ✓ | ✓ |
| WEP-Schlüssel | wesside-ng | ✓ | ✓ |

✓ = ja; × = nein

Tabelle 4.6: Untersuchte Angriffe auf Wireless LANs

Mit Ausnahme des Association- und Authentication-Floodings waren alle Angriffe erfolgreich, d. h. das jeweilige Angriffsziel wurde erreicht. Wahrscheinlich verfügen die verwendeten Access Points über einen proprietären Schutzmechanismus, der sie vor diesen Typen von DoS-Attacken schützt. Zumindest lässt dies eine Recherche in einschlägigen Internet-Foren vermuten. Wichtiger ist aber, dass der vorgestellte Prototyp eines WLAN-IDS alle Angriffe zweifelsfrei als solche erkannt hat.

[6] http://www.vmware.com/ (Abruf: August 2008)
[7] http://www.remote-exploit.org/backtrack.html (Abruf: August 2008)

Dies gilt auch für den zweiten Teil, der Validierung der verteilten Angriffserkennung. Diese erfolgte wie bereits erwähnt in einer emulierten Testumgebung, wobei auf den Einsatz konkreter Angriffswerkzeuge verzichtet wurde, deren grundsätzliche Erkennung ja bereits im ersten Teil nachgewiesen wurde. Stattdessen wurde die Generierung der entsprechenden Ereignisse, etwa beim Eintreffen eines Deauthentication Frames, durch ein eigens entwickeltes Policy-Skript simuliert. Sowohl die gegenseitige Registrierung der kooperierenden Systeme als Observer, wie auch die gemeinsame Erkennung des Angriffs sowie die Verbreitung von Angriffsmeldungen funktionierte problemlos.

## 4.5 Zusammenfassung

In diesem Kapitel wurde die Entwicklung einer Systemlösung zur Angriffserkennung in Wireless LANs vorgestellt. Die hierbei angewandte Methodik orientiert sich an der *Object Modeling Technique* [RBP+93] und lässt sich grundsätzlich auf beliebige Übertragungsstandards anwenden. Sie umfasst die in Abbildung 4.7 dargestellten Schritte:

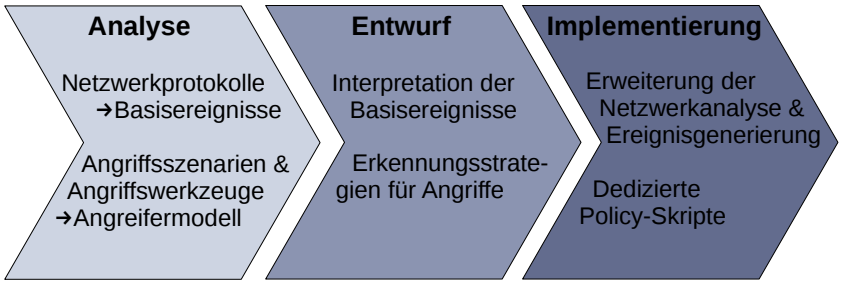



Abbildung 4.7: Angewandte Methodik zur Entwicklung der Wireless IDS-Lösung

Die Ausgangslage bildet einerseits die Analyse der betrachteten Netzwerkprotokolle, um so grundlegende Ereignisse während der Kommunikation zu identifizieren. Je nach Detaillierungsgrad kann dies in der teilweisen oder sogar vollständigen Modellierung des Übertragungsprotokolls in Form endlicher Automaten resultieren. Dabei wird entweder durch den Eintritt eines Zustands oder durch den Übergang von einem Zustand in einen anderen ein Ereignis ausgelöst. In der Praxis dürfte jedoch eine vollständige Darstellung des Protokolls als Zustandsgraph meist nicht nötig sein. So wurden in dieser Arbeit parallel zur Analyse des IEEE 802.11 Standards bekannte Angriffsszenarien und -werkzeuge untersucht. Dabei stellte sich heraus, dass quasi alle Angriffe auf der IEEE 801.11 Management-Ebene durchgeführt werden. Folglich konzentrierten sich die entworfenen Strategien zur Erkennung von Angriffen auf die Interpretation der von Management Frames ausgelösten Ereignisse.

Zusätzlich wird auf externe Informationen zurückgegriffen, wie sie etwa Netzmanagementsysteme bereitstellen. Die entwickelten Strategien erscheinen dabei auf den ersten Blick verblüffend einfach. McHugh hat dazu in [McH01] sehr treffend festgestellt:

> *One of the more frustrating aspects of intrusion detection is that many of the common attacks are enabled by easily avoidable vulnerabilities. The kind of errors that are rampant in commonly used software should not be present at all.*

Tatsächlich sind die Strategien jedoch ausreichend, um den Einsatz herkömmlicher Exploits aufzudecken, wie die durchgeführten Experimente gezeigt haben. Diese lassen aber auch bereits erahnen, dass die Komplexität der Policy-Skripte mit den potentiellen Fähigkeiten des Angreifers wächst.

Auch die Implementierung gliederte sich in zwei Teile. Für die Analyse der empfangenen Netzwerkpakete wurde auf das objektorientierte Bro IDS-Framework zurückgegriffen, das bereits weitreichende Netzanalysefunktionalität mitbringt. Diese setzte jedoch bisher erst auf Schicht 3 des ISO/OSI-Stacks an. Zur Unterstützung von IEEE 802.11 wurde daher die Interpretation der auf Schicht 2 ausgetauschten Frames in einer seperaten Klasse gekapselt und in das bestehende Framework derart integriert, dass die passende Analyseroutine automatisch anhand des auf Schicht 2 übermittelten Netzwerktyps ausgewählt wird. Zur Ergänzung weiterer Übertragungsprotokolle kann analog verfahren werden.

Die Analyse selbst besteht aus der Zerlegung der empfangenen Netzpakete gemäß des Übertragungsstandards sowie der Generierung der während des Analyseschritts festgelegten Basisereignisse. Für diese werden zudem grundlegende *Event Handler* implementiert. Bei der Realisierung der Policy-Skripte zur Erkennung konkreter Angriffsszenarien hat sich gezeigt, dass die Trennung zwischen grundlegender Protokollanalyse und Interpretation der dabei eintretenden Ereignisse vorteilhaft ist, da sich die Strategien zur Erkennung von Angriffen auf diese Weise sehr einfach sukzessive verbessern und an veränderte Rahmenbedingungen anpassen lassen. Dies wurde einerseits am Beispiel der verteilten Erkennung eines Man-in-the-Middle-Angriffs durch enge Kooperationsgruppen demonstriert und wird andererseits durch die im Rahmen mehrerer Laborversuche gesammelten Erfahrungen gestützt.

Die im Rahmen dieser Experimente durchgeführte erste Evaluierung hat die grundsätzliche Eignung des entwickelten Prototypen für die Erkennung von Angriffen auf Wireless LANs bestätigt. Um diesen zu einer praktisch einsetzbaren Systemlösung auszubauen, sind allerdings noch verschiedene Probleme technischer Natur zu lösen. Beispielhaft sei hier die Einteilung des verwendeten Frequenzbands in verschiedene Kanäle genannt. Der verwendete Kanal wird für die Interpretation der erzeugten Ereignisse benötigt. Er wird derzeit aus dem Header des übermittelten WLAN Frames bzw. aus den Einstellungen der für das Monitoring verwendeten Netzwerkkarte ermittelt. Während

ersteres jedoch nicht gefeit ist vor Manipulationen, ist die zweite Methode anfällig für Interferenzen.

Neben der eindeutigen Bestimmung des tatsächlich verwendeten Kanals stellt in der Praxis auch die simultane Überwachung mehrerer Kanäle ein Problem dar. Dies wird von handelsüblicher Hardware nicht unterstützt. Neben speziell entwickelter Hardware bieten hier Techniken zur Virtualisierung von WLAN-Netzwerkschnittstellen, wie etwa in [Cha06] geschildert, oder aber die gemeinsame Überwachung des Frequenzspektrums durch eine Kooperationsgruppe mögliche Lösungsansätze. Virtuelle Netzwerkkarten können darüber hinaus zum Aufbau dedizierter Kommunikationskanäle für die Interaktion in Kooperationsgruppen genutzt werden.

Gravierender als die geschilderten technischen Probleme, da konzeptioneller Natur, erscheinen hingegen die Mängel der Kommunikationsschnittstelle für die verteilte Erkennung. Diese ermöglicht bisher nur eine relativ statische Kopplung einzelner Mobilgeräte, wie sie lediglich in infrastrukturellen WLANs und überschaubaren Ad-hoc-Netzen praktikabel erscheint. Der bisweilen hohen Dynamik der für mobile Ad-hoc-Netze vorgeschlagenen losen Kooperationsgruppen wird sie jedoch nicht gerecht. Hier setzt die im folgenden Kapitel vorgeschlagene Systemlösung zur dynamischen Kopplung von Endgeräten an.

# 5

# Systemlösung zur losen Kopplung mobiler Endgeräte

Zur weiteren Realisierung der in Kapitel 3 beschriebenen Rahmenarchitektur wird nun eine Systemlösung zur dynamischen Kopplung von Endgeräten vorgestellt, um Angriffe gemeinsam zu erkennen. Damit wird die Idee des Zusammenspiels in *einem* verteilten System, das sich beispielsweise aus den Mobilgeräten eines Nutzers zusammensetzt, erweitert zur Zusammenarbeit *mehrerer* voneinander abgrenzbarer Systeme. Zur Anbahnung solcher Kooperationen müssen die drei funktionalen Bestandteile Partnersuche, Partnerauswahl und Koordination der Rahmenarchitektur in die Praxis umgesetzt werden (vgl. Abbildung 3.1 auf Seite 62). Neben ihrem selbstorganisierenden Charakter unterscheiden sich solche lose gekoppelten von den in Kapitel 4.4.4 umgesetzten engen Kooperationsgruppen vor allem in ihrer Vertrauensstruktur, da die beteiligten Endgeräte in der Regel verschiedenen Benutzern gehören.

Das hierzu entwickelte System wird im Folgenden zunächst konzeptuell vorgestellt. Die Darstellung konzentriert sich dabei auf die horizontale Kooperationssicht des allgemeinen HUSAR-Modells (vgl. Abbildung 3.1 auf Seite 62). Dabei werden auch die Auswirkungen mobiler Umgebungen auf wesentliche Kernaspekte des Entwurfs – Peer-to-Peer-Netze, Ortsbezug von Netzknoten und Vertrauensbildung in mobilen Umgebungen – betrachtet. Anschließend wird detailliert auf den konkreten Systementwurf eingegangen, der schließlich mittels einer prototypischen Implementierung validiert wird.

## 5.1 Konzeption

Abbildung 5.1 zeigt die von der Kooperationsumgebung bereitzustellenden Funktionen. Ihr grundsätzlicher Aufgabenbereich lässt sich in drei Punkten zusammenfassen:

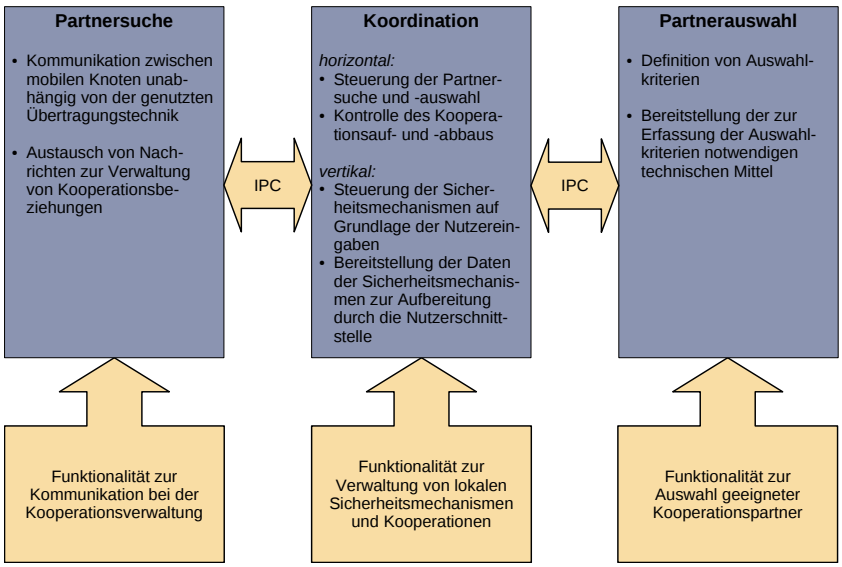



Abbildung 5.1: Konzeption der Kooperations-Middleware für mobile Endgeräte

*Partnersuche:* Die Suche und das Auffinden mobiler Endgeräte, die als mögliche Kooperationspartner fungieren können.

Hierzu ist eine von der genutzten Übertragungstechnik unabhängige Kommunikation zwischen den mobilen Endgeräten notwendig, um den zur Verwaltung der Kooperationsbeziehungen notwendigen Nachrichtenaustausch zu ermöglichen. Um dabei auch infrastrukturlose Netzarchitekturen zu unterstützen, wird für dieses Overlay-Netzwerk auf Peer-to-Peer (P2P) Technologien zurückgegriffen. Kapitel 5.2.1 diskutiert hierzu grundsätzliche Einschränkungen, die sich im mobilen Umfeld ergeben.

*Partnerauswahl:* Die Einschätzung des von einer konkreten Kooperation erwarteten Nutzens sowie die Auswahl der vielversprechendsten Kooperationspartner.

Hierzu sind entsprechende Auswahlkriterien festzulegen und die zu ihrer Erfassung notwendigen technischen Mittel bereitzustellen. Ein mögliches Beispiel eines solchen Auswahlkriteriums ist die räumliche Position eines Knotens. Dies beruht auf der Annahme, dass die Knoten im näheren Umfeld eine ähnliche Sicht auf die eigene Systemumgebung haben und damit besonders wertvolle Hinweise

für die Erkennung von Angriffen liefern können. Darüber hinaus erscheint es einleuchtend, dass ein Knoten hauptsächlich an Ereignissen in seinem direkten Umfeld interessiert ist. Kapitel 5.2.2 umreißt deshalb kurz grundlegende Ansätze zur Lokalisierung mobiler Knoten.

Als weiteres Beispiel eines Auswahlkriteriums dient das in ein bestimmtes Endgerät gesetzte Vertrauen, um den erwarteten Nutzen einer Kooperation einzuschätzen. Der Systementwurf in Kapitel 5.3 definiert daher eine generische Schnittstelle zu einem beliebig gearteten Trust-Framework, um aus einer gegebenen Menge von Endgeräten die vertrauenswürdigsten auszuwählen.

Partnersuche und Partnerauswahl bilden zusammen die sogenannten Kooperationsmechanismen.

*Koordination:* Die Kopplung der separierten Nutzerschnittstelle, der Sicherheits- und der Kooperationsmechanismen.

Dies beinhaltet insbesondere die funktionale und dynamische Modellierung der Kooperationsverwaltung, sowie die Realisierung der Interprozesskommunikation (engl. *Inter-Process Communication – IPC*) zur Kopplung der einzelnen Systemteile. Mit der Trennung von Programmlogik und Nutzerinteraktion folgt der vorgeschlagene Ansatz dem Model-View-Controller-Architekturmuster [Ree79] und erleichtert so eine spätere Erweiterung und Anpassung des entwickelten Prototypen.

Die grundlegenden Eigenschaften mobiler Umgebungen, die zu dem vorgestellten Lösungsansatz führten, wurden bereits in Kapitel 2.1 auf Seite 11 erläutert. Ein wesentliches Merkmal ist dabei ihre Heterogenität sowohl hinsichtlich der eingesetzten Endgeräte als auch der verwendeten Netzarchitekturen. Um auf unterschiedlichen Endgeräten in beliebigen drahtlosen Netzen nutzbar zu sein, ist die vorgeschlagene Lösung aus Netzwerksicht selbstorganisierend und trifft grundsätzlich keinerlei Annahmen über eine möglicherweise nicht erreichbare Infrastruktur. Der Aufbau des peer-to-peer-basierten Overlay-Netzes ermöglicht außerdem die Kooperation über Netzgrenzen hinweg.

Den Unterschieden der einzelnen Geräteklassen im Hinblick auf die Möglichkeiten ihrer Nutzerschnittstellen wird durch deren Isolation Rechnung getragen. Um eine möglichst große Bandbreite an Betriebssystemumgebungen abzudecken, basiert die Systemimplementierung auf der Java-Plattform.

## 5.2 Analyse der Systemumgebung

Vor der Übertragung des allgemeinen Konzepts in einen Systementwurf werden zunächst noch einige grundsätzliche Entscheidungen hinsichtlich der Gestaltung des

Peer-to-Peer-Overlay-Netzes und der Lokalisierung mobiler Knoten getroffen. Diese werden mit den besonderen Eigenschaften mobiler Umgebungen begründet.

### 5.2.1 Peer-to-Peer-Systeme in mobilen Umgebungen

Aufgrund der augenscheinlichen Gemeinsamkeiten drahtloser Multi-Hop-Netze mit P2P-Systemen, erscheint die Verwendung letztgenannter bei der Realisierung dezentraler und verteilter Dienste in mobilen Umgebungen einleuchtend. Beide Technologien sind selbstorganisierend und verfügen weder über zentrale Kontrollinstanzen noch können sie im Vorfeld Aussagen zur Zu- und Abwanderung von Endgeräten machen. Es gibt jedoch auch Unterschiede, die die uneingeschränkte Anwendung existierender P2P-Ansätze im mobilen Umfeld behindern.

So etablieren P2P-Systeme ein Overlay-Netz auf Anwendungsebene. Hinsichtlich dem zugrunde liegenden physischen Netz gehen gängige Ansätze von einer dem Internet vergleichbaren Struktur aus.

Drahtlose Multi-Hop-Netze spezifizieren hingegen die untersten drei Schichten des ISO/OSI-Protokollstapels. Die so gebildete Netzarchitektur mobiler Ad-hoc-Netze unterscheidet sich dabei fundamental von der kabelbasierten Internet-Architektur. Insbesondere erfolgt das Routing auf der Sicherungsschicht. Der Einsatz heutiger P2P-Systeme im mobilen Bereich ohne Berücksichtigung der auf den unterschiedlichen Schichten wirkenden Mechanismen zur Selbstorganisation führt daher in der Regel zu Problemen. Die ineffiziente Nutzung des physischen Netzes durch gängige P2P-Systeme resultiert beispielsweise in Zickzack-Routen [SK03]. Das damit verbundene erhöhte Verkehrsaufkommen zieht bereits in herkömmlichen Netzstrukturen eine deutliche Verschlechterung der Skalierbarkeit bei steigenden Kosten nach sich [RIF02]. Es ist zu erwarten, dass sich dieser Effekt in mobilen Umgebungen noch verstärkt, da in diesen die zur Verfügung stehende Bandbreite eine wesentlich knappere Ressource darstellt. Die Notwendigkeit einer schichtenübergreifenden Interaktion des P2P-Systems mit den Routing-Mechanismen des physischen Netzes wurde zwar erkannt [ESZK04, GLR05, LW06], bisher fehlen jedoch konkrete Vorschläge für solch integrierte Ansätze. Die wenigen existierenden Arbeiten in diesem Bereich, wie etwa [WZS04, Zah06], sind rein theoretischer Natur ohne praxistaugliche Implementierung.

#### Klassifikation von P2P-Technologien

P2P-Systeme lassen sich allgemein anhand ihrer Netztopologie wie in Abbildung 5.2 auf der nächsten Seite dargestellt klassifizieren [ES05].

*Zentralisierte Systeme* realisieren zwar den Datenaustausch individuell, benötigen zum Routing jedoch einen zentralen Server. Auch wenn es sich damit streng genommen um eine Client/Server-Architektur handelt, werden diese Systeme aus historischen Gründen zu den P2P-Systemen gerechnet. Zwar ermöglicht das zentrale Verzeichnis

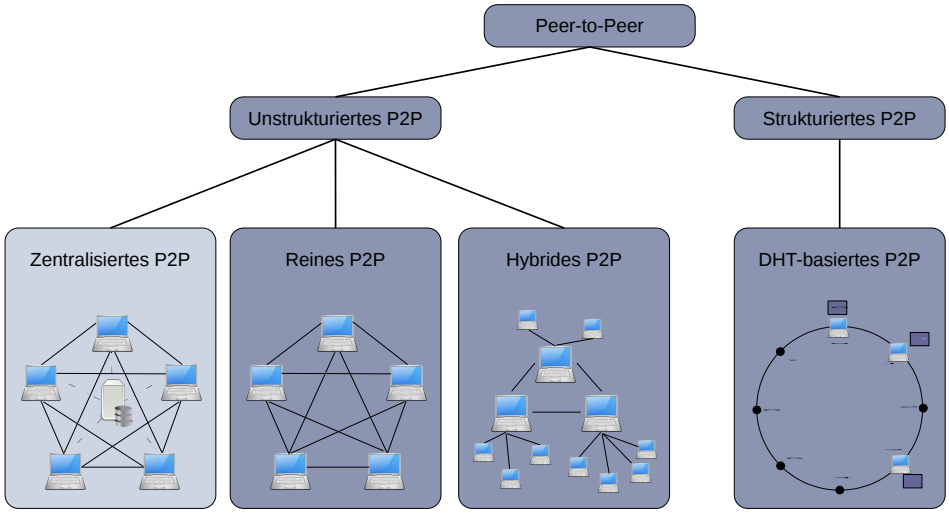



Abbildung 5.2: Klassifikation von P2P-Systemen

eine sehr effiziente Suche nach Ressourcen, es ist dabei jedoch gleichzeitig ein existenzieller Schwachpunkt dieses Ansatzes, der mit zunehmender Anzahl von Peers einen immer engeren Flaschenhals darstellt. Im mobilen Umfeld ist dieser Ansatz zudem nicht praktikabel, da die notwendige Erreichbarkeit des zentralen Verzeichnisdiensts nicht dauerhaft gewährleistet werden kann.

Diesen Nachteil beheben *reine P2P-Systeme*, die auf jegliche zentrale Infrastruktur verzichten. Das Routing erfolgt bei ihnen reaktiv, wobei die hierzu notwendigen Signalisierungsnachrichten durch das Netz geflutet werden. Die dezentrale Organisation wird also mit erhöhtem Kommunikationsaufwand erkauft. Zwar lässt sich die Ausbreitung von Signalisierungsnachrichten begrenzen, jedoch wird dadurch unter Umständen das Auffinden von Ressourcen verhindert. Letztlich haben Untersuchungen wie [Rit01] gezeigt, dass der reine P2P-Ansatz hinsichtlich des Bandbreitenbedarfs nicht skaliert.

Stattdessen werden heute *hybride Systeme* verwendet, die das P2P-Netz dynamisch strukturieren. Hierfür wählen einzelne Knoten einen sogenannten *Ultra-* oder *Super-Peer*, der für diese Gruppe von Knoten als zentraler Verzeichnisdienst fungiert. Die verschiedenen Ultra-Peers bilden wiederum ein gemeinsames P2P-Netz. Das hierarchische Routing koordinieren somit die Ultra-Peers, die sich untereinander austauschen. Die hierzu notwendigen Signalisierungsnachrichten zwischen den einzelnen Peer-Gruppen werden wie in den reinen P2P-Netzen durch das Overlay-Netz der Ultra-Peers propagiert.

Sowohl in zentralisierten, reinen als auch hybriden Systemen verteilen sich die bereitgestellten Ressourcen vollkommen willkürlich auf die einzelnen Peers. Damit erfolgt auch die Bildung des Overlay-Netzes, in dem die untereinander bekannten Peers miteinander verknüpft sind, gänzlich unstrukturiert. Man spricht daher auch von *unstruk-*

*turierten P2P-Systemen*[1]. Die Lokalisation einer gesuchten Ressource erfordert damit bis zu einem gewissen Grad das Fluten von Suchanfragen durch das Overlay-Netz, wobei es durchaus passieren kann, dass eine Ressource „übersehen" wird.

Eine andere Herangehensweise verfolgen *strukturierte Ansätze*, die sowohl die Peers als auch die von ihnen bereitgestellten Ressourcen auf einen einheitlichen Adressraum abbilden. Dafür werden meist Verfahren auf Basis verteilter Hashtabellen (engl. *Distributed Hash Tables – DHT*) verwendet. Im Gegensatz zu den unstrukturierten Ansätzen wird also sowohl die Topologie des Overlay-Netzes als auch die Verteilung der bereitgestellten Ressourcen durch das P2P-Protokoll festgelegt, so dass in jedem Fall auf einen verfügbaren Peer bzw. die von ihm bereitgestellten Ressourcen zugegriffen werden kann. Dieses proaktive Routing erfordert jedoch zusätzlichen Signalisierungsaufwand zur Umverteilung der Ressourcen beim Zu- und Abgang von Peers, wobei insbesondere ungeplante Abgänge von Peers zu berücksichtigen sind.

#### Probleme gängiger P2P-Technologien im Anwendungskontext

Wie bereits eingangs erwähnt, wird die unveränderte Anwendung existierender P2P-Technologien in drahtlosen Netzen allgemein als nicht sinnvoll erachtet. Die hierbei auftretenden Probleme resultieren aus den Eigenschaften mobiler Umgebungen, wie etwa den Beschränkungen hinsichtlich verfügbarer Bandbreite, vor allem aber dem Fehlen einer dedizierten Netzinfrastruktur. So kann die für zentralisierte Systeme erforderliche permanente Erreichbarkeit des Verzeichnisdienstes in mobilen Ad-hoc-Netzen nicht gewährleistet werden.

Ein für alle P2P-Technologien wichtiger Aspekt ist der sogenannte *Overlay Stretch* [WZS04]. Mit der Anzahl physischer Hops im Overlay-Netz im Verhältnis zu der des theoretisch kürzesten Pfades definiert er ein Maß für die Effizienz eines P2P-Systems bei der Bandbreitennutzung. Die geringere verfügbare Bandbreite und der Umstand, dass die Paketweiterleitung von ressourcenbeschränkten mobilen Endgeräten realisiert werden muss, macht eine effiziente Bandbreitennutzung in drahtlosen Netzen noch wichtiger als in kabelbasierten. So reduziert jeder zusätzliche drahtlose Hop die Wahrscheinlichkeit, dass ein Paket sein Ziel erreicht. Bisherige Analysen haben gezeigt, dass strukturierte P2P-Schemata hier deutlich besser abschneiden als unstrukturierte. So liegen etwa die für *Pastry* [RD01] ermittelten Werte von 1,3 bis 2,4 nahe am optimalen Overlay Stretch von 1,0 [WZS04].

DHT-basierte Lösungen eignen sich jedoch aufgrund ihrer proaktiven Routing-Strategie nur für relativ stabile Umgebungen mit mehr oder weniger gleichbleibenden Netztopologien [ESZK04]. Dies mag in zellulären Netzen noch der Fall sein, MANETs

[1] Hierzu sei angemerkt, dass sich der Terminus *unstrukturiertes P2P* in der Literatur durchgesetzt hat (siehe beispielsweise [ES05]) und daher auch hier verwendet wird. Allerdings ist er im Hinblick auf zentralisiertes und hybrides P2P nicht präzise, da diese als über den zentralen Server respektive den Ultra-Peer strukturiert betrachtet werden können.

zeichnen sich jedoch durch eine hohe Dynamik aus, die zu häufigen Veränderungen auf physischer Ebene führen. Vor diesem Hintergrund erscheint die Verwendung strukturierter P2P-Verfahren wenig ratsam, da zudem praxistaugliche Implementierungen fehlen.

Hinsichtlich ihrer Sicherheit stellen sich bei P2P-Systemen ganz ähnliche Fragen wie etwa beim Routing in Ad-hoc-Netzen (vgl. auch Kapitel 2.4.2). Genau wie bei diesen wurde bei der Entwicklung von P2P-Systemen stillschweigend die Bereitschaft der Peers zur Kooperation und zu regelkonformen Verhalten vorausgesetzt. Tatsächlich gibt es aber sowohl auf Anwendungs- als auch auf Netzwerkebene eine Reihe von Angriffsmöglichkeiten [DKA05]. DHT-basierte Systeme haben sich dabei besonders anfällig gegen Insider-Angriffe gezeigt [WGR05]. Erste Untersuchungen unstrukturierter Systeme hingegen belegen deren Anfälligkeit für das Ausspähen der Netztopologie [RIF02] und die Manipulation der Signalisierung [ZY02]. Beides lässt sich für gezielte Denial-of-Service-Angriffe ausnutzen.

#### Fazit

Die Auswahl einer geeigneten P2P-Architektur erweist sich damit als schwierig. Zwar spricht die effizientere Nutzung der zur Verfügung stehenden Bandbreite für den Einsatz eines DHT-basierten Ansatzes, gerade weil es bereits erste Ansätze für eine schichtenübergreifende Abstimmung des physischen mit dem Overlay-Routing gibt [Zah06]. Durch vermehrten Signalisierungsverkehr verliert der proaktive Ansatz jedoch seine Vorteile, sobald es zu häufigen Änderungen der Netzstruktur kommt. Auch fehlt es derzeit an praxistauglichen Implementierungen. In beiden Punkten sind die unstrukturierten Systeme voraus. In punkto Sicherheit wurden beide Ansätze bisher kaum untersucht.

Der im Folgenden beschriebene Systementwurf abstrahiert daher von der P2P-Lösung und betrachtet lediglich die notwendigen Schritte, um die Kommunikation zwischen Endgeräten unabhängig von der darunter liegenden Netztopologie und dem konkret verwendeten Übertragungsprotokoll zu ermöglichen. Die Implementierung verfolgt dann einen pragmatischen Ansatz, indem – ähnlich der in [SK03] vorgestellten Lösung – das Kooperations-Framework auf Grundlage einer hybriden P2P-Architektur realisiert wird, gleichzeitig jedoch das Overlay-Routing durch die extern bestimmten Positionsangaben der Peers optimiert wird.

### 5.2.2 Lokalisation in mobilen Umgebungen

In mobilen Netzen existieren viele unterschiedliche Verfahren zur Positionsbestimmung. Diese lassen sich zum einen danach unterscheiden, ob die Position eines mobilen Endgeräts von diesem selbst (sogenanntes *Positioning*) oder von anderen (sogenanntes *Tracking*)

bestimmt wird. Ferner ist eine Klassifikation anhand der zur Positionsbestimmung verwendeten Basistechniken möglich [Rot05b].

Bei der Auswahl eines konkreten Verfahrens muss die geplante Anwendungsumgebung berücksichtigt werden. So unterscheiden sich die einzelnen Verfahren nicht nur hinsichtlich ihrer Genauigkeit, sondern sie stellen auch unterschiedliche Anforderungen an die Anwendungsumgebung. Satellitengestützte Verfahren, wie das *Global Positioning System (GPS)*, sind beispielsweise innerhalb von Gebäuden nicht ohne weiteres nutzbar. Andere Techniken greifen auf eine bestehende Infrastruktur zurück, sei es ein Sensornetz oder eine Datenbank, um die Position eines Endgeräts aus seinem aktuellen Umgebungskontext herzuleiten. Kritisch zu sehen ist auch die Weitergabe der Position. Durch die weitgehende Personalisierung mobiler Endgeräte werden hierdurch nicht nur gezielte Angriffe auf bestimmte Geräte ermöglicht, sondern darüber hinaus auch eklatante Eingriffe in die Privatsphäre des Nutzers. Unter Berücksichtigung des Aspekts der informationellen Selbstbestimmung weisen daher Positioning-Verfahren klare Vorzüge auf, da hier der Nutzer selbst entscheiden kann, ob, wem und mit welcher Genauigkeit er seine Position preisgibt. Dabei wird natürlich die Annahme getroffen, dass ein Angreifer aus dem gewählten Verfahren und der bei seiner Abwicklung anfallenden Kommunikation keinerlei Rückschlüsse ziehen kann. Für das rein passiv arbeitende GPS ist das in der Regel der Fall.

Dies und die zuletzt stark wachsende Verbreitung entsprechend ausgerüsteter mobiler Endgeräte sprechen für eine kombinierte GPS-basierte Lösung zur Positionsbestimmung. Aus Gründen der Flexibilität abstrahiert der im Folgenden beschriebene Systementwurf jedoch vom konkreten Lokalisationsverfahren und definiert stattdessen eine allgemeine Schnittstelle zur Anbindung eines beliebigen Positioning-Systems, mit dem ein Endgerät seine aktuellen geographischen Koordinaten bestimmen kann. Dies hat auch den Vorteil, die Übergabe korrekter Koordinaten voraussetzen zu können. Die hierfür notwendige Berücksichtigung sicherheitsrelevanter Fragestellungen muss auf Ebene des konkreten Lokalisationssystems erfolgen.

### 5.2.3 Vertrauensbildung in mobilen Umgebungen

Neben der effizienten Nutzung der zur Verfügung stehenden Ressourcen wurde in Kapitel 5.2.1 als weiteres grundlegendes Problem offener verteilter Umgebungen im Allgemeinen und P2P-Netze im Besonderen ihre Sicherheit identifiziert. Hierbei sind im wesentlichen zwei Aspekte zu berücksichtigen:

1. *Sicherheit der Kommunikation:* Wie kann die Vertraulichkeit, Integrität und Verfügbarkeit der ausgetauschten Nachrichten gewährleistet werden?

2. *Sicherheit der Kooperation:* Wie können die in Kapitel 2.3.3 definierten Schutzziele Diskretion, Konformität und Kooperationsbereitschaft für die beteiligten Netzknoten gewährleistet werden?

Die Besonderheiten dezentral organisierter Systeme erschweren dabei den Einsatz von Sicherheitsmechanismen, die entweder grundlegende Kenntnisse der Partner übereinander (z. B. durch den Austausch kryptographischer Schlüssel) oder die Existenz einer vertrauenswürdigen dritten Partei voraussetzen. Stattdessen wird idealistisch von einer altruistisch geprägten Haltung ausgegangen.

Ein solch blindes Vertrauen in das Wohlverhalten jedes Einzelnen ist aber mit wachsender Nutzerzahl immer schwieriger zu rechtfertigen, besonders da die einzelnen Teilnehmer verhältnismäßig anonym miteinander interagieren. Dennoch ist das Konzept des Vertrauens grundsätzlich von Vorteil, wie Engler in [Eng07, Kapitel 2.1.4] zusammenfassend feststellt. So vereinfacht Vertrauen die Modellierung komplexer sozioökonomischer Zusammenhänge und vermindert das bei Interaktionen wahrgenommene Risiko. Beides führt zu einer Reduktion der Transaktionskosten und damit zu einem größeren Handlungsspielraum für alle beteiligten Entitäten. Im Ergebnis lassen sich so Kooperationen und damit die Leistungsfähigkeit der verteilten Systemumgebung verbessern.

Letztlich scheint Vertrauen eine Grundvoraussetzung für das Zusammenleben in jeder uns bekannten Gesellschaft zu sein. Dabei beeinflussen sich Vertrauen und Sicherheit gegenseitig [AR05, Kapitel 1.3]. Einerseits kann Vertrauen fehlende Sicherheit wenigstens teilweise kompensieren, andererseits ist ein Mindestmaß an Sicherheit erforderlich, um Vertrauen zu schaffen. Camp geht sogar soweit, Vertrauen als die Schnittmenge von Security, Privacy und Reliability zu bezeichnen [Cam03]. Dementsprechend existieren eine Vielzahl interdisziplinärer Untersuchungen zu den Auswirkungen und Ursachen von Vertrauen.

Ursprünglich eine Fragestellung von Theologen, Psychologen und Philosophen, lassen sich seit Anfang der 1990er Jahre vermehrt Arbeiten finden, die ökonomische, politische und soziologische Betrachtungen zu den Auswirkungen von Vertrauen anstellen. Mit der zunehmenden Kommerzialisierung des Internets wurde schließlich um die Jahrtausendwende das Potential von Vertrauen auch in immer mehr Teilgebieten der Informatik erfasst. Trotz oder gerade wegen der Vielzahl der daraus resultierenden wissenschaftlichen Publikationen, fehlt bis heute eine einheitliche Definition des Begriffs Vertrauen.

#### Allgemeine Begriffsbestimmung

Der Duden definiert *Vertrauen* als *„festes Überzeugtsein von der Verlässlichkeit, Zuverlässigkeit einer Person, Sache“* [Dud07]. Ganz ähnlich umschreibt das Oxford Dictionary den englischen Begriff *Trust* als *„belief or willingness to believe that one can rely on the goodness, strength, ability, etc of somebody or something“* [Oxf89]. Vertrauen beschreibt also den Glauben an den (erwarteten) positiven Verlauf einer Entwicklung bezogen auf eine Person oder Sache und ermöglicht so auf Intuition gestützte Entscheidungen.

Diese allgemeinen Definitionen werfen zunächst mehr Fragen auf, als sie beantworten, wobei im Hinblick auf die Bildung von Vertrauen zwei Punkte besonders hervorstechen:

1. Welche Faktoren tragen dazu bei, jemanden von der Verlässlichkeit und Zuverlässigkeit einer Person oder Sache zu überzeugen?

2. Unterscheidet sich das Vertrauen in Personen von dem in Sachen?

Die erste Frage war Gegenstand vieler Untersuchungen. Hinsichtlich ihrer Anwendbarkeit in der Informatik ist jedoch die 1994 erschienene strukturwissenschaftliche Arbeit von Marsh [Mar94] besonders erwähnenswert. Ausgehend von humanwissenschaftlichen Betrachtungen formalisiert sie erstmals die Erfassung von Vertrauen in einem mathematischen Modell und ermöglicht damit nicht nur eine bessere Diskussion des Phänomens Vertrauen, sondern auch seine Nutzung zur Konstruktion neuartiger IT-Systeme.

Eine differenziertere Unterscheidung des Vertrauens in Personen von jenem in Sachen liefert Jøsang in seinem 1996 erschienenen Artikel [Jøs96], der letztere (*rational entities*) von ersteren (*passionate entities*) beeinflusst sieht:

> *Trust in a passionate entity is the belief that it will behave without malicious intent . . . Trust in a rational entity is the belief that it will resist malicious manipulation by a passionate entity.*

Diese Defintion bringt zum Ausdruck, dass Personen im Gegensatz zu Sachen selbst entscheiden können, wie sie sich verhalten. Sachen unterliegen dagegen immer dem Einfluss von Personen. Außerdem wird der eingangs erwähnte Zusammenhang zwischen Vertrauen und Sicherheit unterstrichen, da von „böswilligen Manipulationen“ die Rede ist. In der Folge hat sich in der Informatik ein an den Methoden des Risikomanagements orientiertes Vertrauensbild etabliert, das Vertrauen statistisch zu erfassen sucht.

#### Der Vertrauensbegriff in der Informatik

Ziel der Vertrauensbetrachtungen in der Informatik ist die Entwicklung sogenannter *Vertrauensmanagementsysteme* (engl. *Trust Management Systems*) zur Unterstützung vertrauensbasierter Entscheidungen in der digitalen Welt. Die Grundlage jedes Vertrauensmanagementsystems bildet ein *Vertrauensmodell* (engl. *Trust Model*), das die Vertrauen beeinflussenden Faktoren und ihre Wirkungsweise beschreibt. Dieses wird präzisiert durch eine *Vertrauensmetrik* (engl. *Trust Metric*) zur Darstellung des Vertrauens. Die in den verschiedenen Arbeiten vorgeschlagenen Darstellungsformen lassen beliebige Mengen an Vertrauenswerten zu. Darüber hinaus müssen Algorithmen zur initialen Bildung von Vertrauen *(Trust Formation)* und dessen Weiterentwicklung *(Trust Evolution)*

sowie Protokolle zur Verbreitung von Vertrauensinformationen *(Trust Dissemination)* entworfen werden [Cap04].

Einen breitgefächerten Überblick zum aktuellen Stand der Forschung liefern Artz und Gil in [AG07]. Sie diskutieren mehr als hundert Publikationen aus verschiedenen Gebieten der Informatik, die sie drei grundlegenden Abstraktionsebenen zuordnen:

Arbeiten zu *policybasierten Vertrauensmodellen* basieren auf dem Austausch von Beglaubigungen (engl. *Credentials*) zur Durchsetzung von Zugriffsregeln. Sie spiegeln damit die Implementierungsebene wider. Vertrauen in eine Entität wird durch eine hinreichende Anzahl von ihr vorzulegender Beglaubigungen repräsentiert, die die Berechtigung zur Durchführung einer bestimmten Aktion belegen.

Beglaubigungen sind ein gutes Mittel, um initiales Vertrauen bei Interaktionen zwischen zuvor unbekannten Parteien herzustellen. Leider werden sie bei ihrer Präsentation selbst Gegenstand vertrauensbasierter Entscheidungen. Dieses rekursive Vertrauensproblem wird häufig mittels einer vertrauenswürdigen dritten Partei (engl. *Trusted Third Party – TTP*) gelöst, die als Autorität zur Ausstellung und Überprüfung von Beglaubigungen dient. Das Vertrauen in eine solche TTP sowie ihre permanente Verfügbarkeit ist im mobilen Umfeld jedoch nur schwer zu realisieren (vgl. Kapitel 1.2).

Arbeiten zu *reputationsbasierten Vertrauensmodellen* repräsentieren die Entwurfssicht bei der Konzeption eines Vertrauensmanagementsystems. Das zur Durchführung einer Interaktion notwendige Vertrauen wird anhand bereits gesammelter Erfahrungen etabliert. Im Gegensatz zu den policybasierten Vertrauensmodellen berücksichtigen sie damit die zeitliche Entwicklung der Vertrauenswürdigkeit. Hierzu wird eine Historie des Verhaltens einer Entität erstellt, aus der sich dessen Reputation ableiten lässt. Dabei können ausschließlich eigene Erfahrungen oder aber auch die anderer verwendet werden. Zur Beantwortung der hieraus resultierenden rekursiven Vertrauensfrage („Kann ich auf die Erfahrungen eines anderen vertrauen oder nicht?“) werden mehrere Ansätze diskutiert: der implizite Rückgriff auf die Beziehungsgeflechte in sozialen Netzen, die explizite Festlegung von Vertrauenspfaden in einem Netz oder der Rückgriff auf Beglaubigungen.

Arbeiten zur *allgemeinen Vertrauensmodellierung* dienen quasi als Schnittstelle zu anderen Disziplinen und entsprechen damit der Analysesicht bei der Konstruktion von Vertrauensmanagementsystemen. Im Gegensatz zu den ersten beiden eher anwendungsorientierten Feldern handelt es sich dabei vor allem um strukturwissenschaftliche Grundlagenforschung, die beispielsweise in Zusammenarbeit mit der Psychologie und Soziologie zu ergründen versucht, was Vertrauen bedeutet, worauf es sich gründet und wie es sich auswirkt. Kurzum, es sollen die Charakteristika von Vertrauen erfasst und modelliert werden.

**Merkmale von Vertrauen**

Erkenntnisse über die Eigenschaften von Vertrauen sind eine Voraussetzung, um realistische Konzepte der Trust Evolution zu entwickeln. Aus den Vertrauensmerkmalen lassen sich außerdem Anforderungen an die Protokolle der Trust Dissemination ableiten. Genau wie die Definition von Vertrauen bereitet jedoch die Beschreibung seiner Merkmale bisher noch Schwierigkeiten. Dennoch gibt es einige allgemein akzeptierte Charakteristika, die auch Eingang in die Vertrauensmodelle der Informatik gefunden haben [Vil05]:

- *Vertrauen ist subjektiv,* d. h. die Einschätzung von Vertrauen in ein Vertrauensobjekt wird aus der klar umrissenen Perspektive eines Vertrauenssubjekts in einer bestimmten Rolle getroffen.
- *Vertrauen muss nicht global gültig sein,* d. h. es muss keine Entität geben, der jeder vertraut. Dies folgt direkt aus dem gerade genannten Merkmal.
- *Vertrauen ist kontextsensitiv.* Im Gegensatz zu der Subjektivität genannten Begrenzung des Vertrauenssubjekts, dient die Kontextsensitivität der genaueren Spezifikation des Vertrauensobjekts. Als Beispiel sei der KFZ-Mechaniker genannt, dem man zwar eine Autoreparatur, nicht aber einen chirurgischen Eingriff zutraut. Auch spielt die Kosten-Nutzen-Abschätzung des Vertrauensobjekts bei der Vertrauensbewertung eine Rolle.
- *Vertrauen muss nicht symmetrisch sein.* Wenn Alice Bob vertraut, folgt daraus nicht, dass Bob auch Alice vertraut.
- *Vertrauen muss nicht distributiv sein.* Wenn Alice Bob und Carla gemeinsam vertraut, folgt daraus nicht, dass sie beiden einzeln traut. Dieser Umstand bildet die Grundlage für das Vier-Augen-Prinzip.
- *Vertrauen muss nicht transitiv sein,* d. h. aus „Alice vertraut Bob“ und „Bob vertraut Carla“ folgt nicht zwangsläufig, dass Alice auch Carla vertraut.

**Fazit**

Trotz zahlreicher interdisziplinärer Forschungsarbeiten ist die Frage der Bedeutung von Vertrauen und seiner Modellierung immer noch ungelöst. Aufgrund der fehlenden holistischen Betrachtung konzentrieren sich existierende Lösungen zwangsläufig nur auf Teilaspekte. Analysiert man diese vor dem Hintergrund der in Kapitel 1.2 beschriebenen Einschränkungen, kommt man zum Schluss, dass die policybasierten Ansätze wegen ihrer Abhängigkeit von einer vertrauenswürdigen Infrastruktur im mobilen Umfeld nicht ohne weiteres einsetzbar sind. Die Arbeiten zur allgemeinen Vertrauensmodellierung liefern zwar wichtige Anregungen, lassen aber eine systemorientierte Sicht vermissen.

Zur Gewährleistung der Sicherheit von Kooperationen erscheint daher die Verwendung eines reputationsbasierten Vertrauensmodells unverzichtbar. Diese Hypothese wird durch erste Arbeiten im Bereich der P2P-Netze aber auch beim mobilen Ad-hoc-Routing (siehe Kapitel 2.6.2) gestützt. Deshalb werden in Kapitel 6 erste Lösungsvorschläge für den Entwurf eines Reputationssystems für mobile Umgebungen skizziert.

## 5.3 Systementwurf

Der in Abbildung 5.3 dargestellte Systementwurf ist eine direkte Umsetzung der in Kapitel 3 vorgestellten generischen Rahmenarchitektur. In den folgenden Abschnitten werden die Funktionalität der einzelnen Komponenten und ihr Zusammenwirken genauer vorgestellt.

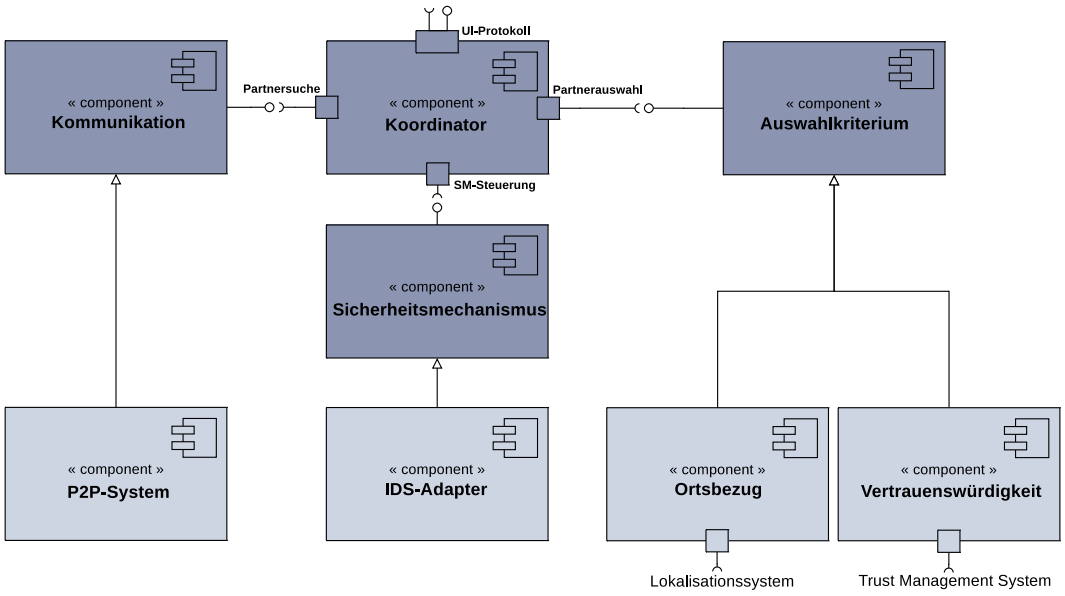



Abbildung 5.3: Komponentendiagramm des Kooperations-Frameworks

### 5.3.1 Koordination von Kooperationen

Bei der Verwaltung von Kooperationen wird im wesentlichen zwischen der Initiierung neuer Kooperationen und der Pflege bereits bestehender unterschieden. Realisiert wird dies durch einen *Koordinator*, der hierfür mit der *Kommunikation* und beliebigen *Auswahlkriterien* zusammenarbeitet. Zu diesem Zweck stellt die Komponente die beiden

Schnittstellen *Partnersuche* und *Partnerauswahl* bereit. Darüber hinaus definiert sie die Schnittstelle *SM-Steuerung* zur lokalen Anbindung eines Sicherheitsmechanismus.

Kern des Koordinators ist ein Verzeichnis aller laufenden Kooperationen, das sogenannte *Kooperationsverzeichnis* $\mathcal{K} = \mathcal{M} \times \mathcal{N} \times \mathcal{T}$, wobei $\mathcal{M}$ die Menge der Sicherheitsmechanismen und $\mathcal{N}$ die Menge aller auf Ebene der Kommunikationskomponente eindeutig und sicher identifizierbaren Knoten bezeichnet, während $\mathcal{T}$ die Gültigkeitsdauer der Kooperationsbeziehungen angibt. Die Aufrechterhaltung einer Kooperation muss dabei regelmäßig bestätigt werden, um konsistente Zustände auch im Fehlerfall zu gewährleisten (*Soft-State-Ansatz*). Beim Aufbau einer Kooperationsbeziehung muss der Koordinator nun dieser Liste einen neuen Eintrag hinzufügen. Bestehende Einträge sind hingegen periodisch auf ihre Gültigkeit zu überprüfen und ggf. zu entfernen. Hierbei ist zu beachten, dass der Koordinator lediglich zwischen den kooperierenden Knoten vermittelt. Die eigentliche Zusammenarbeit findet auf Ebene der Sicherheitsmechanismen statt. Der Koordinator muss diese daher über den Auf- und Abbau einer Kooperation unterrichten. Die Abläufe dieser vom Kommunikationsaufkommen der eigentlichen Kooperation zweier Partner Alice und Bob getrennten Out-of-Band-Signalisierung (siehe Abbildung 5.4) werden im Folgenden genauer vorgestellt.

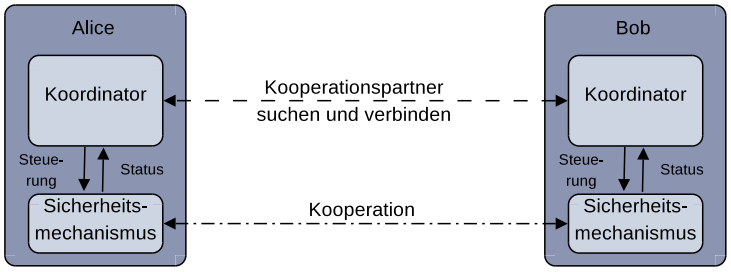



Abbildung 5.4: Kommunikationskanäle zur Realisierung von Kooperationen

### Aufbau neuer Kooperationen

Für jeden kooperativen Sicherheitsmechanismus $m \in \mathcal{M}$ verwaltet der Koordinator einen Wert $k_m^{\max} \in \mathbb{N}$, der die maximale Anzahl der für diesen Mechanismus zulässigen Kooperationen angibt. Dieser Wert wird entweder vom Nutzer vorkonfiguriert oder aber (dynamisch) aus dem Gerätekontext abgeleitet. Wenn nun die Anzahl $k_m \in \mathbb{N}$ der momentan unterhaltenen Kooperationen kleiner ist als $k_m^{\max}$, dann versucht der Koordinator eine neue Kooperation zu initiieren. Der allgemeine Ablauf beim Aufbau neuer Kooperationen wird in Abbildung 5.5 auf der nächsten Seite dargestellt. Er gliedert sich in vier grundlegende Schritte:

1. *Suche potentieller Partner:* Im ersten Schritt werden in der aktuellen Umgebung des mobilen Endgeräts potentielle Kooperationspartner gesucht. Zur Bestimmung

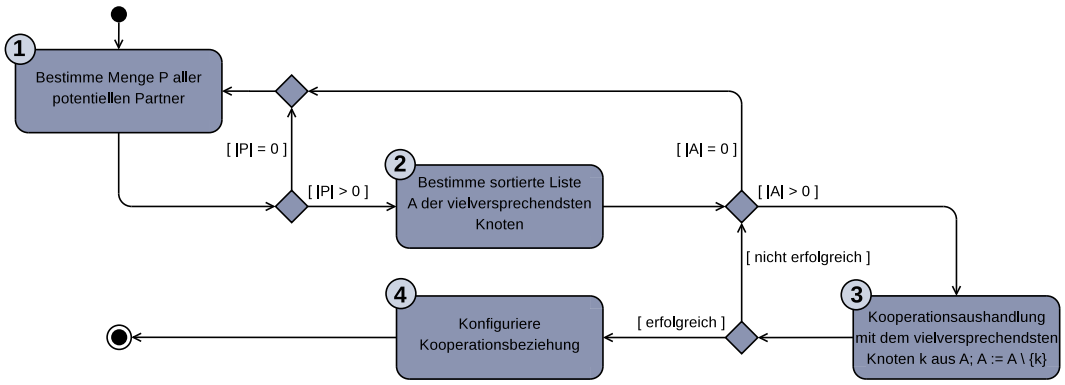



Abbildung 5.5: Aufbau einer neuen Kooperation

dieser Menge $\mathcal{P} \subseteq \mathcal{N}$ stellt der Koordinator über die Kommunikationskomponente eine Suchanfrage. Der genaue Ablauf einer solchen Suche wird durch die Schnittstelle *Partnersuche* spezifiziert. Mittels der Methode $\mathrm{searchFor}\,(m \in \mathcal{M})$ fordert sie die Kommunikationskomponente auf, nach allen mobilen Knoten zu suchen, die den gewünschten kooperativen Sicherheitsmechanismus $m$ unterstützen. Als Ergebnis liefert diese Methode die Menge $\mathcal{P}$ aller potentiellen Kooperationspartner für den Mechanismus $m$ zurück.

2. *Auswahl geeigneter Partner:* Jede Zusammenarbeit ist für die beteiligten Partner mit Aufwand im Sinne von bereitzustellenden Systemressourcen verbunden. Da diese besonders im mobilen Umfeld grundsätzlich knapp bemessen sind, sollte vor dem Eintritt in eine Kooperationsbeziehung der erwartete Nutzen mit dem zu leistenden Aufwand verglichen werden. Nur wenn der Nutzen die notwendigen Aufwände übersteigt, ist eine Kooperation sinnvoll. Andernfalls hätte man besser darauf verzichtet.

   Im zweiten Schritt wird daher die Menge $\mathcal{P}$ der potentiellen Kooperationspartner reduziert und geordnet, indem auf sie verschiedene Kriterien gemäß einer festzulegenden *Auswahlstrategie* angewandt werden. Jedes *Auswahlkriterium* definiert dabei eine Ordinalskala, anhand der eine gegebene Menge mobiler Knoten sortiert wird. Die *Auswahlstrategie* kombiniert wiederum die einzelnen Auswahlkriterien, um eine global sortierte Menge $\mathcal{A} \subseteq \mathcal{P}$ der vielversprechendsten mobilen Knoten zu erhalten, mit denen noch keine Kooperation eingegangen wurde. Es werden aber nicht nur bestehende Kooperationspartner aussortiert. Zusätzlich kann auch jedes Auswahlkriterium einzelne Knoten ausschließen, so dass $|\mathcal{A}| \leq |\mathcal{P}|$ gilt.

   Im Sinne des Strategie-Entwurfsmusters [GHJV95] implementieren die einzelnen Auswahlstrategien somit die abstrakte Schnittstelle *Partnerauswahl*. Diese übergibt eine beliebige Knotenmenge an ein Auswahlkriterium zur Bewertung

und erhält eine sortierte und ggf. reduzierte Knotenmenge zurück. Dies kann für mehrere Kriterien wiederholt werden, wobei die Auswahlstrategie für die Kombination der einzelnen Teilergebnisse verantwortlich ist. Konkretisiert wird dieser Ansatz an zwei Beispielen zur Einbeziehung des Ortsbezugs und der Vertrauenswürdigkeit potentieller Kooperationspartner, wobei die Kombination der Einzelergebnisse durch Schnittmengenbildung erfolgt. In die engere Auswahl kommen also nur diejenigen Knoten, die den gewünschten Sicherheitsmechanismus unterstützen, sich in der näheren Umgebung befinden und vertrauenswürdig sind.

Durch diese rationale Kosten-Nutzen-Abwägung unterscheidet sich der vorgeschlagene Systementwurf von bisherigen altruistisch geprägten Ansätzen. Vor allem die bereits in Kapitel 5.2.3 thematisierte flexible Einbeziehung von Mechanismen zur Bewertung der Vertrauenswürdigkeit mobiler Knoten erscheint für sicherheitskritische Anwendungen unabdingbar.

3. *Aushandeln von Kooperationsbeziehungen:* Bis zu diesem Punkt berücksichtigt die Auswahl der vielversprechendsten Partner lediglich die Sicht eines Partners. Kooperationen sind jedoch bilateral, d. h. sie müssen für beide Seiten gewinnbringend sein.

   Im dritten Schritt wird daher dem vielversprechendsten Knoten $a_1 \in \mathcal{A}$ ein Kooperationsangebot unterbreitet. Hierfür muss ihm über das Kooperations-Framework eine Nachricht gesendet werden, die neben der eigenen Identität $n \in \mathcal{N}$ auch die gewünschte Zeitspanne $t \in \mathcal{T}$ bis zum automatischen Ablauf der Kooperationsbeziehung sowie den Sicherheitsmechanismus $m \in \mathcal{M}$ angibt, bei dem kooperiert werden soll. Der Knoten $a_1$ kann nun selbst entscheiden, ob er dem Kooperationswunsch nachkommen möchte oder nicht und signalisiert dies dem Anfragenden. Im positiven Fall fahren beide Partner mit der lokalen Konfiguration ihrer nun gemeinsamen Kooperationsbeziehung fort, während im Falle einer Ablehnung ein neuerlicher Versuch mit dem nächstbesten Knoten unternommen wird. Wurden alle Kooperationsanfragen abgelehnt, wird erneut mit der Suche nach potentiellen Partnern begonnen, um zwischenzeitliche Änderungen der Systemumgebung zu berücksichtigen.

4. *Konfiguration von Kooperationsbeziehungen:* Hat ein Knoten in einen Kooperationswunsch eingewilligt, muss auf beiden Seiten die neue Kooperation konfiguriert werden. Zunächst ist dazu das Kooperationsverzeichnis zu aktualisieren.

   Seien $a, b \in \mathcal{N}$ die beiden Kooperationspartner, $m \in \mathcal{M}$ der Sicherheitsmechanismus, für den eine Zusammenarbeit vereinbart wurde, und $t_a, t_b \in \mathcal{T}$ die jeweilig gewünschte Gültigkeitsdauer der Kooperationsbeziehung. Zur Konfiguration der Kooperationsbeziehung sind dabei zunächst auf beiden Seiten die Kooperationsverzeichnisse $\mathcal{K}_a$ bzw. $\mathcal{K}_b$ wie folgt zu ergänzen:

$$\mathcal{K}_a := \mathcal{K}_a \cup \{(m, b, \min(t_a, t_b))\}$$
$$\mathcal{K}_b := \mathcal{K}_b \cup \{(m, a, \min(t_a, t_b))\}$$

Die Gültigkeitsdauer als Maß für die angenommene Dynamik der Systemumgebung wird also von dem Partner mit den höheren Anforderungen bestimmt. Ferner wird davon ausgegangen, dass die Art der Kooperation durch die Angabe des Sicherheitsmechanismus eindeutig bestimmt ist. Um innerhalb eines Sicherheitsmechanismus verschiedene Arten der Kooperation zu unterscheiden, kann hierfür ein zusätzlicher Parameter eingeführt werden, der bei der Aushandlung der Kooperationsbeziehung zusammen mit der Angabe des Sicherheitsmechanismus den Kooperationswunsch genauer spezifiziert. Grundsätzlich kann ein solches Tupel aber auch als separater Sicherheitsmechanismus interpretiert werden, so dass der hier verfolgte Ansatz keine Einschränkung der Allgemeingültigkeit darstellt.

Darüber hinaus muss der Koordinator den entsprechenden Sicherheitsmechanismus $m$ auf die neue Kooperation vorbereiten. Hierfür definiert die abstrakte Schnittstelle *SM-Steuerung* eine Methode, mit der dem Sicherheitsmechanismus die Kontaktdaten des Kooperationspartners übermittelt werden, damit dieser die eigentliche Zusammenarbeit aufnehmen kann.

Bevor nun die Aktualisierung bereits bestehender Kooperationsverbindungen erläutert wird, fasst Abbildung 5.6 auf der nächsten Seite den bei der gerade beschriebenen Funktionalität anfallenden Nachrichtenaustausch zwischen Koordinator, Kommunikationskomponente, Auswahlkriterien sowie Sicherheitsmechanismus zusammen. Die angegebenen Markierungen korrespondieren dabei mit der Zustandsnummerierung in Abbildung 5.5 auf Seite 123.

#### Abbau bestehender Kooperationen

Neben dem Aufbau neuer Kooperationen ist der Koordinator auch für die Pflege bereits bestehender verantwortlich. Er hat insbesondere für das ordnungsgemäße Ende einer Kooperation zu sorgen. Abhängig vom Initiator des Kooperationsendes sind dabei folgende Aktionen durchzuführen:

**Abbau durch den Sicherheitsmechanismus:** Signalisiert ein lokaler Sicherheitsmechanismus $m$ dem Koordinator aus welchen Gründen auch immer, dass die Zusammenarbeit mit einem mobilen Knoten $n$ zu beenden sei, so muss dieser den entsprechenden Eintrag aus dem Kooperationsverzeichnis $\mathcal{K}$ entfernen. Ferner ist der Partner $n$ zu informieren und, falls nötig, die lokale Konfiguration des Sicherheitsmechanismus $m$ anzupassen. Die hierfür benötigten Nachrichtentypen wurden bereits im vorangegangenen Abschnitt spezifiziert.

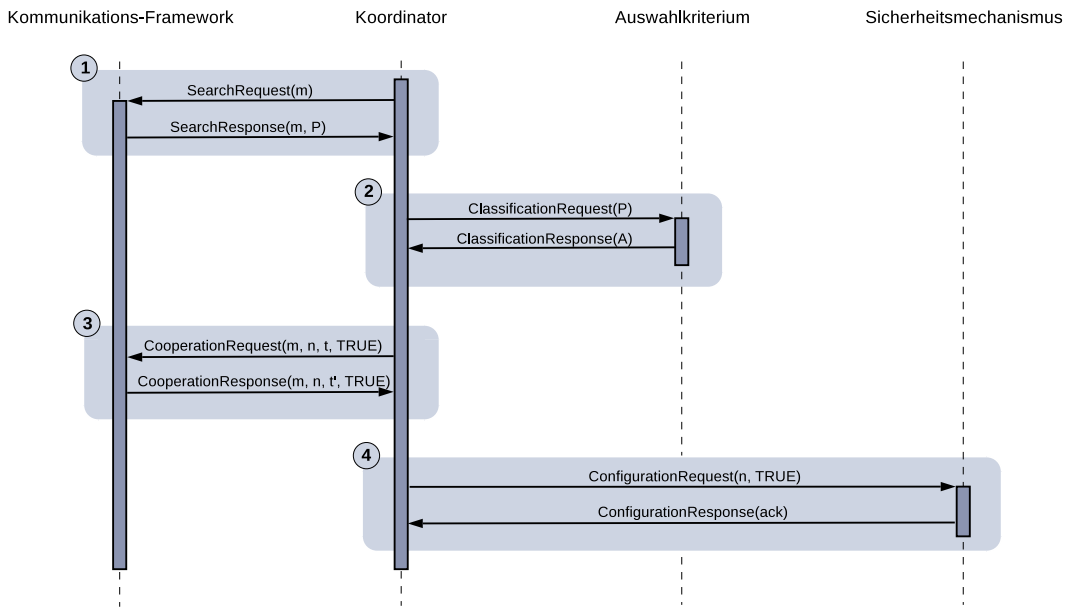



Abbildung 5.6: Kommunikationsflüsse beim Aufbau einer Kooperation

**Abbau durch den Kooperationspartner:** Empfängt der Koordinator über das Kommunikations-Framework die Nachricht eines Partners, dass dieser eine laufende Kooperation aufkündigt, muss er ebenfalls das Kooperationsverzeichnis $\mathcal{K}$ aktualisieren und den betroffenen Sicherheitsmechanismus $m$ hierüber durch Senden eines *Configuration-Request* informieren.

**Abbau durch den Koordinator:** Wie bereits zuvor erwähnt, folgt der Koordinator einem Soft-State-Ansatz, um das System auch bei auftretenden Fehlern konsistent zu halten. In regelmäßigen Abständen müssen daher die Kooperationsbeziehungen erneuert werden. Für die betroffene Beziehung zu einem Knoten $n$ des Kooperationsverzeichnisses sind hierfür die Schritte 2 und 3 gemäß Abbildung 5.5 auf Seite 123 durchzuführen, wobei $\mathcal{P} := \{n\}$. Hierbei wird bewusst auf eine erneute vollständige Suche gemäß Schritt 1 verzichtet. Zwar ließe sich dadurch das Kooperationsverzeichnis optimieren, indem man bestehende Partnerschaften durch höher bewertete ersetzt. Der hierfür zu erbringende Aufwand in Form von Funkkommunikation stünde jedoch in keinem Verhältnis zu der damit verbundenen Nutzensteigerung. Neue Partnerschaften werden damit ausschließlich dann eingegangen, wenn alte explizit aufgekündigt wurden. Dies ist beispielsweise der Fall, wenn die Anwendung der Auswahlstrategie in Schritt 2 zu einer klassifizierten Menge $\mathcal{A} = \emptyset$ führt, der Knoten $n$ sich also nicht mehr für die Zusammenarbeit qualifiziert. Das weitere Vorgehen entspricht dann dem beschriebenen

Kooperationsabbau durch einen Sicherheitsmechanismus. Analog hierzu wird die negative Bestätigung des in Schritt 3 gesendeten *CooperationRequest* als Kündigung durch den Kooperationspartner behandelt.

### 5.3.2 Kommunikation zwischen mobilen Endgeräten

Die Kommunikation zwischen mobilen Endgeräten wird durch die gleichnamige Komponente realisiert. Sie vermittelt damit zwischen den Koordinatoren unterschiedlicher mobiler Knoten und kapselt die Funktionalität des konkreten Overlay-Netzes. Dabei hat sie folgende Aufgabenbereiche abzudecken:

#### Anbindung an die logische Netzstruktur

Bevor ein Knoten mit anderen kooperieren kann, muss er sich zunächst mit der logischen Netzstruktur verbinden. Abhängig von der verwendeten Technologie sind dabei unterschiedliche Vorgehensweisen denkbar. Wie bereits eingangs erläutert, bietet sich für das betrachtete Anwendungsszenario grundsätzlich der Einsatz von P2P-Technologien an. Unstrukturierte Ansätze realisieren den Netzeintritt beispielsweise durch (teilweises) Fluten des Netzes mit speziellen Anfragen. Strukturierte Ansätze erfordern die Aktualisierung der im gesamten Netz verteilten Routing-Informationen, um den neuen Knoten zu berücksichtigen. Unter Umständen muss auch eine Beschreibung des angebotenen Sicherheitsmechanismus veröffentlicht werden, damit der Knoten für andere auffindbar wird.

Der Eintritt in das Overlay-Netz wird ebenfalls durch den Koordinator gesteuert, der hierfür ein *AdvertisementRequest* an die Kommunikationskomponente sendet. Dieser ist mit einer Spezifikation des Sicherheitsmechanismus parametrisiert, für den Kooperationen eingegangen werden sollen. Abhängig von der konkret verwendeten Overlay-Technologie wird diese Beschreibung entweder weitergeleitet oder zur Beantwortung zukünftiger Suchanfragen externer Knoten in der Kommunikationskomponente gepuffert.

#### Suche nach Kooperationspartnern

Die für eine Suche nach neuen Kooperationspartnern notwendigen Informationen übermittelt der Koordinator der Kommunikationskomponente in einem *SearchRequest* (vgl. Abbildung 5.6 auf der vorherigen Seite). Dabei handelt es sich im wesentlichen um eine Spezifikation des betreffenden Sicherheitsmechanismus. Die Kommunikationskomponente leitet diese Anfrage an das Overlay-Netz weiter und nutzt dazu die von diesem bereitgestellte Funktionalität.

#### Vermittlung von Kooperationswünschen

Wurde ein potentieller Partner ausgewählt, so müssen die Modalitäten der angestrebten Kooperation ausgehandelt werden. Der Koordinator signalisiert dies durch einen *CooperationRequest*, bei dem er den fraglichen Sicherheitsmechanismus $m$, den anvisierten Partnerknoten $n$ sowie die angestrebte Kooperationsdauer $t$ angibt (vgl. wiederum Abbildung 5.6 auf Seite 126). Die gleiche Nachricht signalisiert auch das Ende einer Kooperation. Zur Unterscheidung der beiden Fälle dient der Parameter $type \in \mathbb{B}$. Dieser wird zum Kooperationsaufbau auf TRUE und zum -abbau auf FALSE gesetzt.

Die Kommunikationskomponente sendet unter Ausnutzung der vom Overlay-Netz bereitgestellten Funktionalität eine entsprechende Anfrage an den betreffenden Knoten, dessen Koordinator entscheidet, ob er dem Kooperationswunsch stattgibt. Ein über das Overlay-Netz übermittelter positiver Bescheid wird dem Koordinator durch die Generierung einer *CooperationResponse* mit $ack = \text{TRUE}$ signalisiert. Um dem entfernten Knoten das Senden einer negativen Antwort zu ersparen und außerdem zwischenzeitlich aus der Kommunikationsreichweite gefallene Knoten zu berücksichtigen, übermittelt die Kommunikationskomponente nach einer gewissen Zeitspanne automatisch eine negative *CooperationResponse* an den Koordinator.

### 5.3.3 Selektion von Kooperationspartnern

Der mit einer Kooperation verbundene Ressourcenaufwand macht eine Kosten-Nutzen-Abschätzung zur Auswahl der potentesten Partner erforderlich. Zur Bewertung können, wie bereits in Abschnitt 5.3.1 geschildert, beliebige Kriterien definiert werden. Diese lassen sich nach frei zu definierenden Strategien miteinander kombinieren, um eine Auswahl aus den grundsätzlich zur Verfügung stehenden Kooperationspartnern zu treffen. Die Umsetzung der Partnerauswahl durch Anwendung des Strategie-Entwurfsmusters wird in Abbildung 5.7 verdeutlicht.

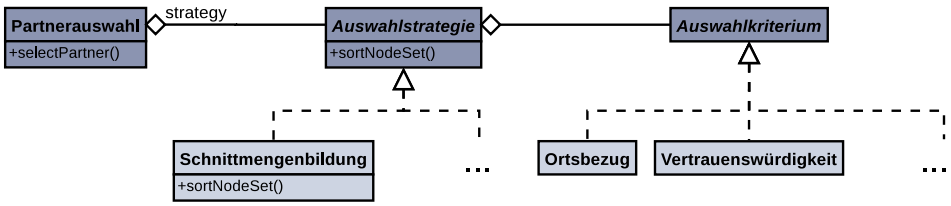



Abbildung 5.7: Realisierung der Partnerwahl

Die Schnittstelle *Partnerauswahl* des Koordinators verwendet die in der abstrakten Klasse *Auswahlstrategie* definierten Operationen, um zunächst eine übergebene Menge mobiler Knoten hinsichtlich beliebiger *Auswahlkriterien* zu sortieren. Jede Sortierung entspricht damit einer *partiellen ordinalen Nutzenquantifizierung*. Zur Definition einer *totalen*

*ordinalen Nutzenfunktion* sind die Ergebnisse der partiellen Nutzenfunktionen geeignet zu kombinieren. Diese Konkretisierung der abstrakten Auswahlstrategie wird exemplarisch anhand einer einfachen *Schnittmengenbildung* demonstriert. Grundsätzlich sind hierfür allerdings beliebig komplexe Algorithmen denkbar. Aus systemtechnischer Sicht fungiert ein Auswahlkriterium außerdem als Adapter zu externen Informationsquellen, wie etwa einem Lokalisations- oder Trust Management System.

### 5.3.4 Integration von Sicherheitsmechanismen

Die Komponente *Sicherheitsmechanismus* repräsentiert einen Adapter zur Einbindung separater Sicherheitssysteme, wie etwa des im vorangegangenen Kapitel vorgestellten Wireless IDS. Hierzu deklariert sie abstrakte Methoden, um zwischen der Koordinatorschnittstelle *SM-Steuerung* und der jeweiligen Nutzungsschnittstelle einer bestimmten Gattung von Sicherheitsmechanismen zu vermitteln. Diese Methoden sind für jeden konkret angebundenen Mechanismus zu implementieren.

Neben der notwendigen Funktionalität zur Realisierung der horizontalen Kooperationsperspektive gemäß des in Abbildung 3.1 auf Seite 62 dargestellten allgemeinen HUSAR-Modells, ist dabei auch die vertikale Sicht der Komposition lokaler Sicherheitsmechanismen zu einem Gesamtsystem zu berücksichtigen. Hierfür muss insbesondere die Möglichkeit geschaffen werden, den aktuellen Status des Sicherheitsmechanismus an den Koordinator zu signalisieren, indem wiederum auf die Mechanismen der Interprozesskommunikation zurückgegriffen wird.

### 5.3.5 Realisierung der Interprozesskommunikation

Die Umsetzung des Datenaustauschs zwischen den einzelnen nebenläufigen Komponenten des Kooperations-Frameworks erfolgt unter Anwendung des *Actor-Modells* [HBS73]. Tabelle 5.1 auf der nächsten Seite fasst die in den vorangegangenen Abschnitten eingeführten Nachrichtentypen zur Verwaltung loser Kooperationen zusammen. In Reaktion auf den Empfang einer Nachricht kann eine Komponente lokale Aktionen ausführen, weitere Komponenten kontaktieren und Informationen an den Absender der auslösenden Nachricht übermitteln. So wird beispielsweise durch das Senden von $ack$ = TRUE bzw. $ack$ = FALSE die erfolgreiche respektive erfolglose Bearbeitung einer Anfrage signalisiert.

Zur Organisation der Nachrichtenübermittlung werden zwischen allen interagierenden Komponenten unidirektionale, asynchrone Verbindungen in Form von *Warteschlangen* (engl. *Message Queues*) eingerichtet. Jede Komponente stellt für alle Komponenten, mit denen sie interagiert, je einen FIFO-Puffer für ein- und ausgehende Nachrichten bereit, um diese bis zu ihrer endgültigen Verarbeitung zwischenzuspeichern. Um in einer Antwort auf eine vorangegangene Anfrage referenzieren zu können, werden sämtliche Nachrichten mit einem eindeutigen Identifikator versehen.

| Nachrichtentyp | Nachrichteninhalte | Erläuterung |
|---|---|---|
| *Koordinator ↔ Kommunikation* | | |
| SearchRequest | $m \in \mathcal{M}$ | Suche Partner für $m$ |
| SearchResponse | $m \in \mathcal{M}, \mathcal{P} \subseteq \mathcal{N}$ | Potentielle Partner $\mathcal{P}$ für $m$ |
| CooperationRequest | $m \in \mathcal{M}, n \in \mathcal{N}, t \in \mathcal{T}, type \in \mathbb{B}$ | Bitte $n$ um Auf-/Abbau der Kooperation bei $m$ |
| CooperationResponse | $m \in \mathcal{M}, n \in \mathcal{N}, t' \in \mathcal{T}, ack \in \mathbb{B}$ | Quittiere $n$'s Kooperationsänderung für $m$ |
| AdvertisementRequest | $m \in \mathcal{M}, type \in \mathbb{B}$ | grundsätzliche Bereitschaft zur Kooperation bei $m$ signalisieren |
| AdvertisementResponse | $ack \in \mathbb{B}$ | Quittiere Anzeige der Kooperationsbereitschaft |
| *Koordinator ↔ Auswahlkriterium* | | |
| ClassificationRequest | $\mathcal{P} \subseteq \mathcal{N}$ | Übergebe Knotenmenge zur Sortierung |
| ClassificationResponse | $\mathcal{A} \subseteq \mathcal{P}$ | Liefere sortierte Knotenmenge |
| *Koordinator ↔ Sicherheitsmechanismus* | | |
| ConfigurationRequest | $n \in \mathcal{N}, t \in \mathcal{T}, type \in \mathbb{B}$ | Hinzufügen/Entfernen eines Partners für $m$ |
| ConfigurationResponse | $ack \in \mathbb{B}$ | Quittiere die Änderung eines Partners |
| RegistrationRequest | $m \in \mathcal{M}, type \in \mathbb{B}$ | Registrierung eines Sicherheitsmechanismus |
| RegistrationResponse | $ack \in \mathbb{B}$ | Quittieren der Registrierung |

Tabelle 5.1: Interprozesskommunikation der Kooperations-Middleware

## 5.4 Prototypische Implementierung

Der soeben beschriebene Systementwurf wurde im Rahmen einer Diplomarbeit implementiert und validiert [Rei07]. Der dabei entstandene Prototyp des *Koordinators* konzentriert sich auf die Integration des im letzten Kapitel vorgestellten Wireless IDS sowie dessen Einbindung in lose Kooperationsgruppen. Die Schnittstellen zur Partnersuche und -auswahl wurden ebenfalls implementiert, wie auch die Komponente *Kommunikation*. Allerdings werden sowohl das Lokalisations- als auch das Trust Management System zur Umsetzung der Auswahlkriterien derzeit nur emuliert.

Im Folgenden wird zunächst die Implementierung der Interprozesskommunikation als Grundlage für die Interaktion der einzelnen Komponenten beschrieben, bevor genauer auf die Erstellung der anderen Systemteile eingegangen wird. Die gesamte

Implementierung des Kooperations-Frameworks erfolgte in der Programmiersprache Java, um einen plattformübergreifenden Einsatz zu vereinfachen.

### 5.4.1 Umsetzung der Interprozesskommunikation

Abbildung 5.8 zeigt die Klassenhierarchie zur Umsetzung der in Tabelle 5.1 auf der vorherigen Seite zusammengefassten Nachrichtentypen für die Koordination von Kooperationen.

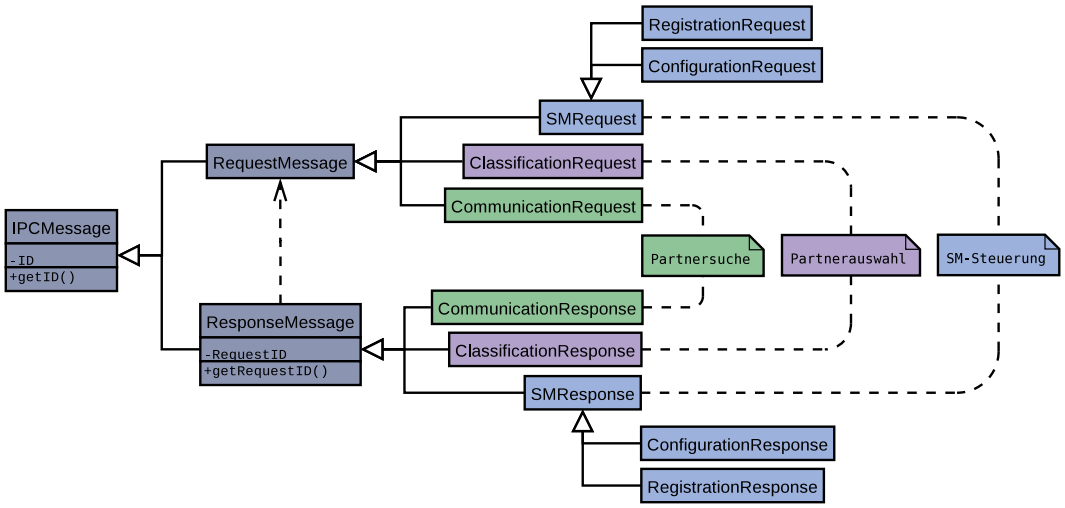



Abbildung 5.8: Implementierung der Interprozesskommunikation

Als Start- bzw. Endpunkt der unidirektionalen Verbindung zwischen zwei Komponenten dienen dabei die Java-Klassen queueOut und queueIn, die beide von der im Java-Paket `java.util.concurrent` definierten Klasse LinkedBlockingDeque abgeleitet werden und somit threadsichere FIFO-Puffer darstellen. Die so realisierte Warteschlange kann damit als universeller Wrapper zur Kopplung beliebiger Dienste an den *Koordinator* genutzt werden, indem sich dieser als *Listener* bei der Ausgangswarteschlange des entsprechenden Diensts registriert. Eine neue Komponente muss dazu lediglich die öffentlich bereitgestellten Methoden addMessageQueueIn(), getMessageQueueOut(), attachListenerQueueOut() und detachListenerQueueOut() implementieren.

### 5.4.2 Umsetzung des Koordinators

Der Kern des *Koordinators* wird in nur einer Java-Klasse implementiert. Dieser CooperationManager kapselt die gesamte Funktionalität zur vollständigen Initialisierung des Systems nebst allen zugrundeliegenden Subsystemen und zur Verwaltung von Koope-

rationspartnern. Der grundlegende Programmablauf wird in Listing 5.1 vereinfacht dargestellt.

```
while ( true ) {
    synchronized ( EventQueue ) {
        while ( EventQueue.isEmpty() ) {
            try {
                EventQueue.wait(); }
            catch ( InterruptedException ex ) {}
        }
        EventSource = EventQueue.pollFirst();
    }
    switch ( EventSource ) {
        case SECURITY_MECHANISM:
            processSecMecMessage( SecMecManager.getResultQueueOut() );
        break;
        case COMMUNICATION:
            processCommMessage( CommManager.getResultQueueOut() );
        break;
    }
}
```

Listing 5.1: Ereignisverarbeitung durch den CooperationManager

Nach dem Programmstart wird anhand der Warteschlange EventQueue permanent überprüft, ob irgendeine Komponente eine Zustandsänderung anzeigt, die eine Interaktion mit dem Koordinator erforderlich macht (Zeilen 3–7). Ist dies der Fall, wird das entsprechende Ereignis aus der Warteschlange entfernt (Zeile 8) und wie in Kapitel 5.3.1 beschrieben verarbeitet, indem die entsprechende Routine zur Auswertung der neuen Nachricht aufgerufen wird (Zeilen 10-17).

Derzeit werden an dieser Stelle lediglich die Komponenten *Kommunikation* und *Sicherheitsmechanismus* unterstützt. Bei einem Ausbau des Prototypen sind hier entsprechende Routinen zur Behandlung weiterer Ereignistypen zu ergänzen.

### 5.4.3 Umsetzung der Partnersuche

Die Suche nach Kooperationspartnern und die Kommunikation mit anderen mobilen Endgeräten mittels peer-to-peer-basierter Mechanismen erfolgt in der Komponente *P2P-System*, die hierfür die P2P-Plattform *JXTA*[2] verwendet.

[2] http://www.jxta.org/ (Abruf: August 2008)

### Grundlagen JXTA

Das maßgeblich von Sun Microsystems vorangetriebene JXTA-Framework definiert die in Abbildung 5.9 dargestellte dreischichtige Software-Architektur zur Entwicklung interoperabler und plattformunabhängiger P2P-Anwendungen [SM07].

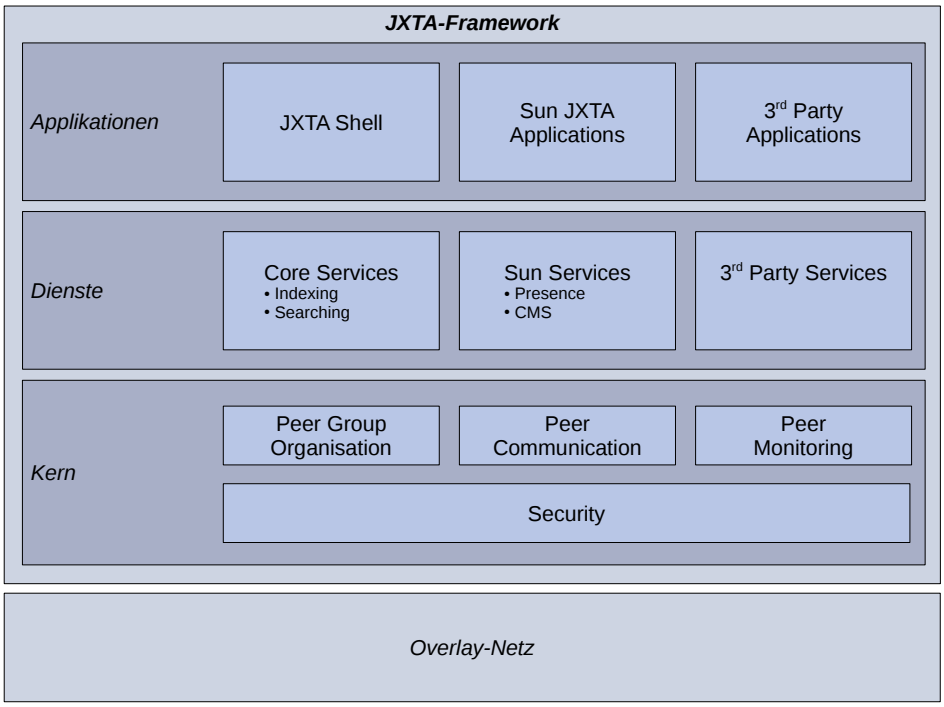



Abbildung 5.9: Die JXTA-Architektur

Im Kern stellt JXTA eine Sammlung offener Protokolle zur Verfügung, um den Informationsaustausch und den Aufbau von Kooperationen zwischen beliebigen vernetzten Endgeräten in P2P-basierten Anwendungen zu standardisieren. Der hierzu notwendige Nachrichtenaustausch basiert auf XML-Dokumenten. Die spezifizierten Protokolle sind grundsätzlich unabhängig von der zugrundeliegenden Netztopologie, dem eingesetzten Transportprotokoll sowie dem Betriebssystem des genutzten Endgeräts. Es existieren Referenzbibliotheken für verschiedene Programmiersprachen, wie etwa Java (Standard und Micro Edition), C, C++ und C#.

Die auf jedem teilnehmenden Peer zu realisierende Kernfunktionalität umfasst das Auffinden anderer Peers und der von ihnen bereitgestellten Ressourcen, die Kommunikation untereinander sowie die Organisation des Overlay-Netzes. Diese ist in den Komponenten *Peer Group Organisation* und *Peer Communication* gekapselt. Darüber

hinaus sind Mechanismen zur Durchsetzung und Überprüfung von Sicherheits- und Verfügbarkeitsanforderungen vorgesehen (Komponenten *Security* und *Peer Monitoring*).

Auf Grundlage dieser Kernfunktionalität stellen (Gruppen von) Peers übergeordnete Dienste und Anwendungen zur Verfügung. Die Unterscheidung zwischen Diensten und Anwendungen ist dabei fließend. Die Systemarchitektur unterscheidet jedoch zwischen Basisdiensten und -anwendungen und solchen, die von Sun Microsystems oder anderen Anbietern zur Verfügung gestellt werden.

#### Implementierung

Die Komponente *P2P-System* hat im wesentlichen zwei Aufgaben zu erfüllen: die *Suche* nach potentiellen Partnern sowie die *Kommunikation* zum Aufbau neuer Kooperationen und ihrer Verwaltung. Sie verwendet hierfür die JXTA-Referenzimplementierung JXSE für die Java Plattform, Standard Edition.

Die Kernfunktionalität von *P2P-System* wird durch die Klasse P2PManager implementiert. Sie realisiert sowohl die Schnittstelle *Partnersuche* zum CooperationManager des *Koordinators* als auch den Zugriff auf das Overlay-Netz.

Für die gemeinsame Angriffserkennung wird initial eine öffentlich zugängliche Peer-Gruppe eingerichtet und über den JXTA-Verzeichnisdienst bekannt gemacht. In dieser müssen sich alle Peers registrieren, die grundsätzlich zur Kooperation bereit sind. Beide Schritte verwenden die Funktionalität der JXTA-Komponente *Peer Group Organisation*. Damit können sich potentielle Partner gegenseitig finden und mittels der in der JXTA-Komponente *Peer Communication* bereitgestellten Funktionalität neue Kooperationen aushandeln.

### 5.4.4 Umsetzung der Partnerauswahl

Der Prototyp implementiert derzeit als einzige Auswahlstrategie die bereits in Kapitel 5.3.3 erläuterte Schnittmengenbildung. Diese ist integraler Bestandteil der zentralen Koordinatorklasse CooperationManager. Als Auswahlkriterien dienen der Ortsbezug sowie die Vertrauenswürdigkeit eines mobilen Netzknotens. Die erforderlichen Schnittstellen zur Anbindung eines Lokalisations- bzw. Trust Management Systems sind in den Klassen PositionSource und TrustSource gekapselt. PositionSource und TrustSource realisieren damit die beiden genannten Auswahlkriterien. Beide werden von der abstrakten Klasse SelectionCriteria abgeleitet, in der die Methoden zum Sortieren gegebener Knotenmengen für ein beliebiges Auswahlkriterium implementiert sind.

PositionSource ermittelt die durch ein Objekt der Klasse Position repräsentierten geographischen Koordinaten eines Knotens und implementiert die notwendigen Methoden zur Bestimmung seiner gegenwärtigen Entfernung. Analog hierzu nutzt die Klasse TrustSystem Objekte der Klasse TrustValue zur Bestimmung der Vertrauenswürdigkeit eines mobilen Knotens.

Auf die Anbindung konkreter Lokalisations- und Trust Management Systeme wurde bisher verzichtet. Der Prototyp emuliert stattdessen frei wählbare Bewegungs- und Vertrauensprofile. Damit lassen sich die Auswirkungen von Änderungen in der Netz- und Vertrauensstruktur einer mobilen Umgebung auf das Kooperations-Framework untersuchen.

### 5.4.5 Anbindung des Wireless IDS

Abbildung 5.10 zeigt die Anbindung eines Sicherheitsmechanismus an das Kooperations-Framework am Beispiel des in Kapitel 4 vorgestellten Wireless IDS. Dessen Integration erfolgt dabei in zwei Schritten.

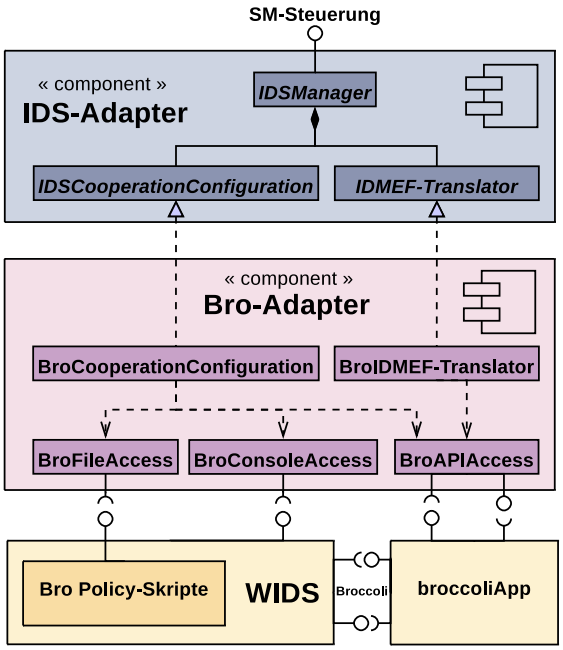



Abbildung 5.10: Integration des Bro-Frameworks

Zunächst wird durch den allgemeinen *IDS-Adapter* eine abstrakte Schnittstelle für die Integration beliebiger IDS in das Kooperations-Framework festgelegt. Diese muss, wie bereits im Systementwurf in Kapitel 5.3.4 erläutert, im wesentlichen zwei Aufgaben erfüllen:

1. Die Konfiguration der an der verteilten Angriffserkennung beteiligten Knoten nach den Vorgaben des *Koordinators*. Die hierfür notwendige Funktionalität wird in der abstrakten Klasse IDSCooperationConfiguration beschrieben.

2. Die Umwandlung der Alarm- und Statusmeldungen des IDS in das standardisierte *Intrusion Detection Message Exchange Format* [DCF06] für eine vereinheitlichte Signalisierung an den *Koordinator*. Die hierbei verwendeten Methoden werden in der Klasse IDMEF-Translator deklariert.

Die Programmlogik der übergeordneten Interaktion zwischen CooperationManager und IDS, d.h. die Umsetzung der Koordinatorschnittstelle *SM-Steuerung*, ist in der Klasse IDSManager gekapselt. Damit ist die allgemeine Implementierungssicht auf die Integration eines IDS in das Kooperations-Framework vollständig beschrieben.

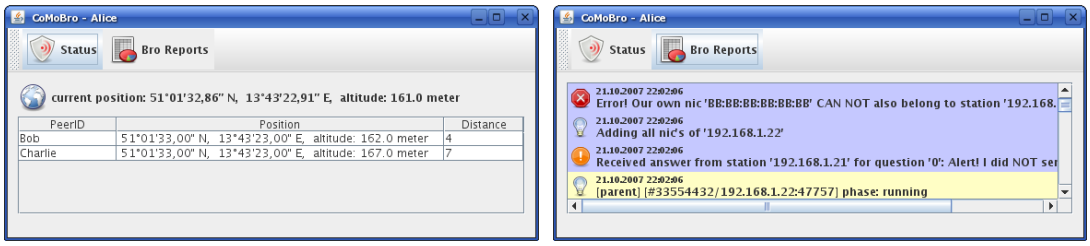
Der solchermaßen definierte abstrakte *IDS-Adapter* wurde in der vorliegenden Prototypimplementierung für das Bro IDS-Framework weiter konkretisiert. So stellen die Klassen BroCooperationConfiguration und BroIDMEF-Translator die bro-spezifischen Funktionen zur Konfiguration der verteilten Angriffserkennung bzw. zur Umwandlung von Bro-Ereignismeldungen bereit. Die Anpassung der in Bro auf Policy-Ebene festgelegten Kooperationen durch BroCooperationConfiguration kann dabei auf zwei Arten erfolgen:

- Für den direkten *Zugriff auf die Policy-Skripte* stellt die Klasse BroFileAccess einen rudimentären Editor bereit, um die Konfiguration der verteilten Angriffserkennung in den entsprechenden Konfigurationsdateien zu ändern. Zur Aktivierung der vorgenommenen Modifikationen sind darüber hinaus die Methoden der Klasse BroConsoleAccess zu verwenden, um die laufende Bro-Instanz neu zu starten. Werden hierbei keine zusätzlichen Vorkehrungen getroffen, führt dies zum vollständigen Verlust des aktuellen Zustands einer laufenden Bro-Instanz, der sich negativ auf die Erkennungsleistung auswirken kann. Zwar werden in [SP05] Mechanismen zum Erhalt des Zustands einer Bro-Instanz entwickelt, die auch teilweise von der vorliegenden Implementierung unterstützt werden. Standardmäßig verfolgt der Prototyp jedoch den im Folgenden beschriebenen Ansatz.
- So ermöglicht die *Nutzung der Programmierschnittstelle Broccoli* die direkte und unterbrechungsfreie Beeinflussung einer laufenden Bro-Instanz bei Erhalt ihres derzeitigen Zustands. Die hierzu benötigten Methoden werden in der Klasse BroAPIAccess realisiert, die über die in C++ implementierte Wrapper-Applikation `broccoliApp` an das laufende Bro angebunden ist. Diese wandelt die Methodenaufrufe von BroAPIAccess in zuvor auf Bro Policy-Ebene definierte und mit entsprechenden Bro Event Handlern unterlegte Ereignisse um und übermittelt diese an das WIDS. Umgekehrt erfolgt über `broccoliApp` auch der Empfang von Status- und Alarmmeldungen des WIDS, die dann zur weiteren Aufbereitung an die Klasse BroIDMEF-Translator weitergereicht werden.

### 5.4.6 Realisierung einer rudimentären Nutzerschnittstelle

Anstatt der vollständigen Spezifikation und Implementierung eines nachrichtenbasierten UI-Protokolls, wurde in dem vorliegenden Prototyp lediglich eine einfache grafische

Benutzeroberfläche in den Koordinator integriert. Sie berücksichtigt sowohl die Kontrolle der eingegangenen Kooperationen als auch der Sicherheitsmechanismen, indem sie dem Anwender eine grundlegende Übersicht seiner Kooperationspartner sowie der vom WIDS gemeldeten Ereignisse zeigt (vgl. Abbildung 5.11a bzw. 5.11b). Auf die Bereitstellung weitergehender Funktionen zur Konfiguration und Beeinflussung einzelner Komponenten wurde verzichtet.




(a) Verwaltung von Kooperationspartnern (b) Anzeige des aktuellen Sicherheitsstatus

Abbildung 5.11: Prototypische Nutzerschnittstelle des Kooperations-Frameworks

### 5.4.7 Evaluation des Prototypen

Ziel der Evaluierung des erstellten Prototypen war die Überprüfung der Funktionalität zum dynamischen Aufbau neuer Kooperationsgruppen im Laborumfeld. Dies umfasst zunächst die Initiierung neuer Kooperationsgruppen, also die Partnersuche und -auswahl sowie die Aushandlung notwendiger Kooperationsparameter. Dies spiegelt die horizontale Sicht des HUSAR-Modells wider. Ferner war das Zusammenspiel von Kooperationskoordinator und Wireless IDS gemäß der vertikalen Sicht des HUSAR-Modells zu überprüfen.

Zu diesem Zweck wurde auf die bereits zur Validierung der verteilten Angriffserkennung in Kapitel 4.4.5 eingesetzte Testumgebung zurückgegriffen, die auf der Virtualisierungssoftware VMware basiert. In der Testumgebung wurden mehrere virtuelle Instanzen des vorgestellten Prototypen emuliert.

Der Aufbau neuer Kooperationen und ihre Darstellung in der rudimentären Nutzeroberfläche funktionierte problemlos, wie Abbildung 5.11a zeigt. Zur Überprüfung der Anbindung des WIDS an das Kooperations-Framework wurde – wie bereits bei der Überprüfung der verteilten Angriffserkennung in Kapitel 4.4.5 – entsprechender Netzverkehr simuliert, so dass auch die lose Kopplung von WIDS erfolgreich getestet werden konnte. Der dritte Listeneintrag in Abbildung 5.11b demonstriert dies, indem er eine Nachricht des WIDS zeigt, die auf ein von einem Kooperationspartner empfangenes Ereignis zurückzuführen ist.

## 5.5 Zusammenfassung

Mit der vorgestellten Systemlösung zur Kopplung von Sicherheitsmechanismen und Endgeräten wurde die prototypische Systemumgebung zur Unterstützung reaktiver Sicherheit in mobilen Netzen komplettiert. Die in diesem Kapitel vorgestellte Kooperations-Middleware ermöglicht die lose Kopplung mobiler Endgeräte über ein Overlay-Netz zur Realisierung kooperativer Sicherheitsmechanismen.

Die Analyse der zugrunde liegenden Systemumgebung identifiziert dabei die effiziente Organisation des Overlay-Netzes als grundlegende Forschungsfrage, die vor einem Einsatz der vorgeschlagenen Systemlösung in der Praxis zu klären ist. Neben einer ortsbezogenen Abbildung des Overlay-Netzes auf die physische Netztopologie gehört hierzu vor allem die Berücksichtigung von Sicherheitsanforderungen. Dies erfordert insbesondere Mechanismen zur Etablierung dynamischer Vertrauensstrukturen. Die Behandlung beider Punkte auf Ebene des Overlay-Netzes ist Gegenstand aktueller Forschung. Solange hier praxistaugliche Lösungen noch fehlen, kann diesen Aspekten allerdings auch auf Applikationsebene Rechnung getragen werden, wie dies der vorgestellte Systementwurf in Form zweier Auswahlkriterien tut.

Der objektorientierte Systementwurf ist modular gestaltet und macht regen Gebrauch von Entwurfsmustern. So sind etwa die zentralen Komponenten *Kommunikation*, *Sicherheitsmechanismus* und *Auswahlkriterium* als Adapter zur Anbindung darunterliegender Systeme konzipiert, während die Interaktion mit dem zentralen *Koordinator* durch den asynchronen Austausch von Nachrichten realisiert wird, die von jeder Komponente vor der Verarbeitung in speziellen Warteschlangen zwischengespeichert werden. Damit gestattet der Entwurf nicht nur einen flexiblen Ausbau des Systems, sondern auch den Austausch einzelner Komponenten, so dass die Systemlösung als Grundlage für den Aufbau beliebiger kooperativer Dienste in mobilen Umgebungen genutzt werden kann. Von besonderem Interesse wäre hier sicherlich eine Erweiterung um kooperative Reaktionen auf erkannte Angriffe.

# 6
# Verlässlichkeit mobiler Umgebungen

In Kapitel 5.2.3 wurde bereits erläutert, warum das häufig angenommene uneigennützige Verhalten mobiler Knoten für konkrete Kooperationsszenarien wenig realistisch erscheint. So gibt es starke Anreize für einen Knoten, sich egoistisch zu verhalten (vgl. Kapitel 2.2.1). Deshalb erscheint es sinnvoller, nicht blind in die Sicherheit einer Kooperationsbeziehung zu vertrauen, sondern diese Entscheidung bewusst zu treffen. Hierfür werden Verfahren und Mechanismen benötigt, mit denen sich die Vertrauenswürdigkeit eines Kooperationspartners einschätzen lässt. Ein wichtiger Anhaltspunkt dafür kann die empfundene Verlässlichkeit eines Partners sein.

Im folgenden wird zunächst die Problemstellung, nämlich die Gewährleistung der Sicherheit von Kooperationen, weiter präzisiert. Dazu wird der in Kapitel 2.3.3 auf Seite 28 eingeführte Begriff der Verlässlichkeit eines Kooperationspartners aufgegriffen und seine Reputation als Maß hierfür definiert. Unter Berücksichtigung der Eigenheiten mobiler Umgebungen werden dann zwei Entwürfe eines Reputationssystems skizziert, um die Erfassung der Reputation technisch zu unterstützen. Das Kapitel schließt mit einer kurzen zusammenfassenden Diskussion.

## 6.1 Problemstellung und Begriffsbestimmung

Die Sicherheit kooperativer Systeme in offenen Umgebungen, wie beispielsweise mobilen Ad-hoc-Netzen, setzt sich aus zwei Aspekten zusammen. Zum einen ist die für verteilte Systeme notwendige Datenübertragung zu schützen (Sicherheit der Kommunikationsebene). Dies kann durch herkömmliche kryptographische Mechanismen realisiert

werden. Andererseits muss aber auch die Sicherheit der Kooperation selbst gewährleistet werden, um eine unautorisierte Informationsveränderung oder -gewinnung vor bzw. nach der Datenübertragung durch einen der Kooperationspartner zu verhindern. Dabei stellt sich die Frage der Vertrauenswürdigkeit eines Partners, die jedoch bei spontanen Zusammenkünften wesentlich schwerer zu beantworten ist als in organisierten Umgebungen. In Kapitel 2.3.3 wurde deshalb die Berücksichtigung der Verlässlichkeit eines mobilen Knotens als Anhaltspunkt für seine Vertrauenswürdigkeit gefordert (Anforderung /R7/ auf Seite 28).

In Anlehnung an die herkömmlichen Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit definiert sich die Verlässlichkeit eines Kooperationspartners aus den drei Bereichen Diskretion, Konformität und Kooperationsbereitschaft. Die Einhaltung aller drei Schutzziele unterliegt dabei letztlich der Einflussname durch den Nutzer und kann daher nicht präventiv durchgesetzt werden, da sich Nutzer nicht unbedingt rational verhalten. Als Ausweg bietet sich die Bewertung der Verlässlichkeit eines Partners am Ende einer Kooperation an, um eine Grundlage für zukünftige Vertrauensentscheidungen zu schaffen.

### 6.1.1 Reputation als Maß der Verlässlichkeit

Reputation bezeichnet den Ruf respektive das Ansehen eines Objekts und ist im Deutschen meist positiv besetzt [Dud07]. Im Englischen beschreibt der Begriff generell die einem Objekt zugesprochenen Fähigkeiten und Eigenschaften: *„what is generally said or believed about the abilities, qualities, etc“* [Oxf89]. Die Reputation eines Objekts basiert damit auf den mit ihm gemachten Erfahrungen: Hat es sich wie erwartet verhalten? Infolgedessen ist die Reputation fest mit der Identität des betrachteten Reputationsobjekts verknüpft. Sie hängt aber auch von der Perspektive des Betrachters ab.

Die grundsätzlichen Auswirkungen solch feedbackbasierter Reputation wurden bereits in [KLR05] beschrieben. Angewandt auf Kooperationen, lassen sich diese folgendermaßen formulieren:

*Selektionseffekt:* Die Bewertungshistorie eines mobilen Knotens erlaubt es anderen, das Risiko einer Kooperation mit ihm einzuschätzen. Abhängig von der persönlichen Risikoscheu können so diejenigen Partner ausgewählt werden, die den eigenen Erwartungen am besten gerecht zu werden versprechen, d. h. am verlässlichsten erscheinen.

*Sanktionierungseffekt:* Um möglichen Nachteilen bei der Selektion von Kooperationspartnern zu entgehen, wird ein Knoten daher versuchen, negative Bewertungen zu vemeiden.

Damit wird auch deutlich, dass die Wirkungsweise von Reputation über die von Bezahlmodellen hinausgeht. Letztere bieten lediglich eine Anreizorientierung. Dadurch

können zwar die in Kapitel 2.2.1 beschriebenen egoistischen Angriffe verhindert werden, indem sie teurer und wenn möglich unrentabel gemacht werden. Reputation ermöglicht darüber hinaus jedoch eine Reaktion auf fehlerhafte oder böswillige Knoten.

Im Hinblick auf die Sicherheit von Kooperationen ist ein mobiler Knoten das Reputationsobjekt, dessen Verhalten bewertet wird. Die Verlässlichkeit dieses Knotens, also die Bewertung, ob er sich diskret, konform und kooperationsbereit verhalten hat, repräsentiert die an ihn gestellten Erwartungen[1]. Diese Bewertungen können von den Kooperationspartnern mit Hilfe eines Reputationssystems gesammelt und zu einem Reputationswert aggregiert werden, der die Auswahl neuer Kooperationspartner beeinflusst.

### Alternative Reputationsmodelle

Die Anzahl der Veröffentlichungen im Bereich Vertrauen und Reputation hat in den vergangenen Jahren geradezu explosionsartig zugenommen. Dies liegt sicherlich an den vielfältigen Anwendungsmöglichkeiten infolge neuer internetbasierter Technologien. Die vorgeschlagenen Lösungen sind dabei meist sehr eng mit einem bestimmten Anwendungsgebiet verknüpft. Aufgrund der Fülle der existierenden Arbeiten konzentriert sich der im Folgenden gegebene Überblick auf die wichtigsten Ansätze. Für detailliertere Darstellungen sei auf [SS05, MGM06, JIB07, RKK07, BMB08] verwiesen.

Sabater und Sierra interpretieren Reputation als Teil des in einem sozialen Netzwerk vorhandenen Wissens [SS05]. Zur weiteren Klassifizierung unterscheiden sie dabei einerseits nach der Herkunft dieses Wissens (engl. *Reputation Type*), andererseits nach seiner Beschaffenheit (engl. *Reputation Nature*).

So kann sich Reputation auf direkten Erfahrungen (engl. *Direct Experiences*) gründen, die man selbst in Interaktionen oder durch Beobachtungen gemacht hat. Sie kann aber auch indirekt durch Informationen anderer gebildet werden (engl. *Indirect Information*). In existierenden Reputationssystemen weniger verbreitet ist die Verwendung der Beziehungsgeflechte in sozialen Netzen und daraus abgeleiteter Informationen zur Bestimmung der Reputation eines Netzknotens. Gleiches gilt für die in den Humanwissenschaften durchaus diskutierte Rolle von Vorurteilen.

Hinsichtlich der Natur des betrachteten Reputationsobjekts finden sich in der Literatur folgende Formen von Reputation:

*Einzelreputation* (engl. *Individual Reputation*), d. h. die Reputation von Personen. Mit Bezug auf die Quelle der reputationsbildenden Informationen kann diese weiter in *direkte* und *indirekte Einzelreputation* unterteilt werden [Mui03].

*Gruppenreputation* (engl. *Group Reputation*) oder *Rollenreputation* bezeichnet das Ansehen einer Gruppe von Personen. Ein alltägliches Beispiel für eine solche Gruppe sind

[1] Es wird also nur die Qualität der Kooperationsumstände (die Kooperationssicherheit) bewertet, nicht die der Kooperationsinhalte (der Nutzen der Zusammenarbeit).

Pfarrer oder Ärzte, in deren Diskretion aufgrund ihrer beruflichen Schweigepflicht in der Regel vertraut wird.

*Produktreputation* (engl. *Product Reputation*) betrachtet im Gegensatz zu den beiden vorangegangenen Formen nicht Personen als Reputationsobjekt, sondern Sachen. Als solches bezeichnet sie den Ruf eines Produkts oder einer Dienstleistung. Ein Beispiel sind die Testberichte unabhängiger Prüfinstitute, wie der Stiftung Warentest.

*Ortsreputation* (engl. *Location Reputation*) gibt die Reputation geographischer Orte an. Ein Beispiel hierfür sind die Empfehlungen in einem Reiseführer.

#### Weitere Anwendungsgebiete

Die Einsatzfelder von Reputationssystemen decken alle Arten von selbstorganisierenden Netzen ab und reichen von Internet-Communities, wie z. B. Online-Foren oder Internet-Marktplätzen, und sozialen Netzen über P2P-Netze bis hin zu mobilen Netzen der vierten Generation.

**Internet-Marktplätze** Aufgrund der ökonomischen Relevanz stammen die meisten der Arbeiten bisher aus dem Bereich der Internet-Marktplätze. So wurden am Beispiel des Branchenprimus eBay eine Vielzahl von Untersuchungen über den Zusammenhang der Reputation des Verkäufers eines Produkts und des dafür erzielten Preises vorgenommen. Die hierbei erzielten Ergebnisse wurden bereits in [Del03] zusammengefasst und sollen deshalb an dieser Stelle nicht wiederholt werden.

**P2P-File-Sharing-Systeme** Als zweites großes Anwendungsfeld von Reputation haben sich in der Informatik P2P-Netze etabliert. Die meisten Arbeiten betrachten hierbei die weit verbreiteten File-Sharing-Systeme. Ihre Ergebnisse lassen sich aber grundsätzlich auf beliebige P2P-Systeme anwenden. Durch die Verwendung von Reputation soll im wesentlichen die Fairness der Nutzung des P2P-Overlay-Netzes gesichert werden. Dabei kommen zwei Aspekte zum Tragen: einerseits die Integrität der angebotenen Daten [CDV+02] und zum anderen ihre Verfügbarkeit [AMCB04, FC05].

**Mobile Umgebungen** Bei der Anwendung von Reputationssystemen in mobilen Umgebungen kann zwischen Lösungsansätzen für die Netzwerk- und solchen für die Anwendungsschicht unterschieden werden. Für die erstgenannte Gruppe sei hier nochmals auf die in Kapitel 2.6.2 genannten Arbeiten zum sicheren Ad-hoc-Routing verwiesen.

Als Beispiel für die Arbeiten auf Applikationsebene sei auf das im Rahmen des iClouds-Projekts[2] konzipierte datenschutzfreundliche Reputationssystem verwiesen,

[2] http://iClouds.tk.informatik.tu-darmstadt.de/ (Abruf: September 2008)

das in [VHM05] skizziert wird. Ziel von iClouds ist die spontane Vernetzung mobiler Endgeräten mit geringer Funkreichweite, um einen peer-to-peer-artigen Austausch beliebiger Informationen zu ermöglichen. Der hierfür entwickelte Prototyp realisiert eine auf IEEE 802.11 basierende infrastrukturelle Single-Hop-Netzarchitektur. Das entworfene Reputationssystem dient dabei zur Beurteilung der Qualität der angebotenen Informationen. Für das Sammeln und Verbreiten der hierzu notwendigen Daten nutzt es die in der iClouds-Architektur *Information Sprinkler* genannten Access Points, die über einen Backbone miteinander gekoppelt sind.

### 6.1.2 Reputationssysteme

Reputationssysteme sammeln, verteilen und aggregieren Feedback über das Verhalten von Kooperationsteilnehmern und ermöglichen so Vertrauensentscheidungen über zuvor Unbekannte zu treffen [RZFK00]. Ihre Funktionalität lässt sich nach [Vos04] wie in Abbildung 6.1 dargestellt modellieren.

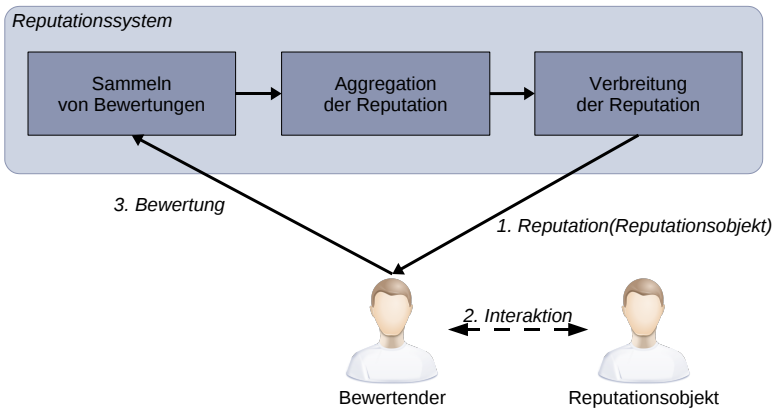



Abbildung 6.1: Funktionales Modell eines Reputationssystems

Als erstes müssen die technischen Mittel bereitgestellt werden, um Bewertungen zu sammeln. Hierfür muss zunächst der Kontext einer Bewertung und ein dazu passender Bewertungsmaßstab festgelegt werden. Auf Grundlage der abgegebenen Bewertungen muss das Reputationssystem dann die einzelnen Bewertungen mittels eines Reputationsalgorithmus zu einer Reputation aggregieren. Agiert das Reputationsobjekt in mehreren Kontexten, können hierbei verschiedene Reputationsalgorithmen zum Einsatz kommen. Die Gesamtreputation des Reputationsobjekts wird in diesem Fall durch einen Vektor repräsentiert. Schließlich sind die Reputationsinformationen entweder auf Verlangen oder automatisch zu verteilen.

### Systemarchitekturen

Zur Realisierung eines solchen Reputationssystems sind unterschiedliche monolithische und verteilte Systemarchitekturen denkbar. Mui et al. kategorisieren diese nach dem verwendeten Reputationsmodell [MMH02]. Alternativ schlägt Voss eine systemorientierte Unterteilung vor, die verschiedene Ansätze nach dem Speicherort der Reputationsinformationen unterscheidet [Vos04]:

*Zentralisierte Reputationssysteme* speichern und verarbeiten die Reputationsdaten auf einem oder mehreren zentralen Servern.

*Subjektive Reputationssysteme* erwägen ausschließlich selbst gemachte Erfahrungen, die jeder Beteiligte in seinen bisherigen Interaktionen mit anderen Teilnehmern gemacht hat. Damit ermöglichen sie nur Vertrauensentscheidungen innerhalb einer relativ stabilen Gruppe untereinander bekannter Knoten.

*Verteilte Reputationssysteme* beheben diesen Nachteil, indem sie zusätzlich die Erfahrungen anderer in die Reputationsberechnung einfließen lassen. Hierfür muss aber nicht nur der aufwändige Austausch von Erfahrungswerten realisiert, sondern auch die daraus resultierende Vertrauensfrage nach ihrer Glaubwürdigkeit beantwortet werden.

In *Reputationssystemen mit lokaler Speicherung* ist jeder Teilnehmer selbst für die Speicherung seiner Reputationsinformationen verantwortlich. Dazu muss er in der Lage sein nachzuweisen, dass er eine Bewertung zu Recht bekommen hat. Ferner muss sichergestellt werden, dass er diese nicht manipuliert hat.

Zwar ist Voss' Einteilung keine richtige Klassifikation, da Kombinationen der einzelnen Formen möglich sind. Jedoch weist sie – wie im Folgenden noch deutlich werden wird – bei der Wahl einer Systemarchitektur zur Bewertung der Verlässlichkeit mobiler Knoten einige Vorteile auf. Sie wird daher trotz ihrer Defizite als Grundlage für die weiteren Betrachtungen herangezogen. Zuvor wird aber noch ein kurzer Überblick über mögliche Berechnungsverfahren für Reputationswerte und die an Reputationssysteme gestellten Sicherheitsanforderungen gegeben.

### Reputationsalgorithmen

Die Kernfunktionalität eines Reputationssystems, die Aggregation der Bewertungen zu einer Reputation, wird durch einen Reputationsalgorithmus umgesetzt. Dieser wird durch das Reputationsmodell vorgegeben. So vielfältig wie die Arbeiten zur Modellierung von Reputation, so unterschiedlich sind auch die vorgeschlagenen Verfahren zur Berechnung von Reputationswerten [JIB07]:

**Summen- oder Durchschnittsbildung:** Die einfachste Form zur Berechnung von Reputationswerten ist die getrennte Summenbildung positiver und negativer Wertungen. Die Differenz aus beiden Werten bildet dann die Gesamtwertung. Eine Verfeinerung dieser Summenbildung ist die Berechnung eines (gewichteten) Mittelwerts aller Bewertungen. Als Wichtungsfaktor kann dazu bespielsweise die aus dem sozialen Netz ermittelte Vertrauensstruktur oder das Alter der Bewertung dienen.

**Bayessche Systeme:** Bayessche Systeme berechnen die statistische Aktualisierung der Beta-Wahrscheinlichkeitsdichtefunktion, um die Reputationswerte aus binären Bewertungen (d. h. positiv oder negativ) zu ermitteln. Ein Reputationswert wird durch das Parametertupel $(\alpha, \beta)$ oder als Erwartungswert der Beta-Wahrscheinlichkeitsdichtefunktion repräsentiert. Dabei gibt $\alpha$ die Anzahl der positiven und $\beta$ die der negativen Bewertungen an.

**Diskrete Vertrauensmodelle:** Anstatt einer kontinuierlichen Bewertungsskala bevorzugen Menschen zur Bewertung häufig diskrete verbale Aussagen, um ihre Einschätzung eines Reputationsobjekts zu formulieren. Ein solch diskretes Vertrauensmodell lässt sich beispielsweise mittels der gebräuchlichen Schulnoten („sehr gut“, „gut“, „befriedigend“ usw.) ausdrücken.

**Belief-Modelle:** Ziel der Belief-Theorie ist die Modellierung der Ungenauigkeit oder Ungewissheit von Informationen. Sie ist eng verbunden mit der Wahrscheinlichkeitstheorie, unterscheidet sich von dieser jedoch darin, dass die Summe aller Eintrittswahrscheinlichkeiten einer Zufallsvariable nicht zwangsläufig 1 ergeben muss. Stattdessen wird die verbleibende Restwahrscheinlichkeit als Ungewissheit interpretiert. Die Belief-Theorie kann zur Ermittlung von Reputationswerten basierend auf den bisherigen Bewertungen eines Reputationsobjekts verwendet werden. Dabei werden nur die Bewertungen von solchen Bewertenden berücksichtigt, für die sich eine transitive Vertrauenskette konstruieren lässt, deren abgeleiteter Vertrauenswert einen vorgegebenen Schwellwert übersteigt.

**Fuzzy-Modelle:** Vertrauen und Reputation kann auch als linguistisch unscharfes Konstrukt aufgefasst werden, bei dem Mitgliedschaftsfunktionen beschreiben, bis zu welchem Grad ein Reputationsobjekt als vertrauenswürdig oder nicht angesehen wird. Als Erweiterung der klassischen, booleschen Logik stellt Fuzzylogik Regeln zum Schlussfolgern mit solch unscharfen Maßen zur Verfügung.

**Flussmodelle:** Reputationswerte, die als Ergebnis transitiver Iterationen durch eine verschlungene oder beliebig lange Kette von Mitgliedern ermittelt wurden, werden

Flussmodell genannt. Manche Flussmodelle setzen konstante Reputationsmengen voraus, so dass ein Teilnehmer seine Reputation nur auf Kosten anderer verbessern kann. Andere Modelle hingegen ermöglichen auch die Neubildung von Reputation.

Aus Sicht des Systementwurfs sollte die Realisierung der Aggregation von Bewertungen modular gestaltet sein, so dass ein beliebiger Reputationsalgorithmus verwendet werden kann. Im folgenden wird daher bewusst von dem konkreten Reputationsalgorithmus abstrahiert. Im Einklang mit den Besonderheiten mobiler Ad-hoc-Netze, in denen im Extremfall nur der Kontakt zu einem direkten Kooperationspartner möglich ist, wird lediglich gefordert, dass der Reputationsalgorithmus alle benötigten Eingabedaten von diesem erhält. Ferner muss der Algorithmus lokal ausgeführt werden, was letztlich auch den subjektiven Charakter von Reputation widerspiegelt.

#### Sicherheitsanforderungen

Als Grundlage von sicherheitsrelevanten Entscheidungen sind Reputationssysteme ein verlockendes Ziel für möglichen Missbrauch. In der Vergangenheit wurden an diversen Stellen der Literatur potentielle Angriffsszenarien und mögliche Gegenmaßnahmen diskutiert. Diese Arbeiten werden in [CH07] erstmals systematisch zusammengefasst. Die daraus abgeleiteten Sicherheitsanforderungen an ein Reputationssystem sind jedoch äußerst vage formuliert. Insbesondere wurde keine Einordnung hinsichtlich der klassischen Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit vorgenommen. Ferner fehlt ein Bezug zu den einzelnen Komponenten eines Reputationssystems sowie eine Unterscheidung der möglicherweise gegenläufigen Interessen von Reputationsobjekt und Bewertendem.

Eine präzisere Untersuchung im Sinne einer mehrseitigen Sicherheitsanalyse liefert Steinbrecher in ihrer Dissertation [Ste08, Kapitel 3.4]. In Anlehnung an das funktionale Systemmodell eines Reputationssystems trennt sie zunächst zwischen der Abgabe und der Auswertung einer Reputation. Für beides formulierte sie Vertraulichkeits-, Integritäts- und Verfügbarkeitsanforderungen. Diese ordnet sie den einzelnen Beteiligten zu, wobei sie zwischen zulässigen Nutzern des Reputationssystems (Bewertender und Reputationsobjekt), dem Reputationssystem selbst und Außenstehenden unterscheidet.

## 6.2 Entwurfsoptionen für ein mobiles Reputationssystem

Die Wahl eines Reputationsmodells sowie die Systemarchitektur des Reputationssystems zu seiner Umsetzung werden beide durch die Eigenschaften des anvisierten Anwendungsgebiets beeinflusst. Für die in dieser Arbeit betrachteten mobilen Umgebungen wurde bereits in Kapitel 2.1.2 festgestellt, dass die verwendeten drahtlosen Netze im

Extremfall lediglich die Verbindung zu *einem* anderen mobilen Knoten ermöglichen (infrastrukturloses Single-Hop-Netz). Setzt man dies in Bezug zu der in Kapitel 6.1.2 vorgestellten Unterteilung von Systemarchitekturen eines Reputationssystems nach Voss, kommt man zu folgenden Schlussfolgerungen:

- *Zentralisierte Lösungen* setzen die permanente Verfügbarkeit eines zentralen Reputationsservers voraus. Dies kann jedoch in infrastrukturlosen drahtlosen Netzen nicht gewährleistet werden, weshalb dieser Ansatz nicht in Frage kommt.
- *Subjektive Ansätze* ermöglichen nur Vertrauensentscheidungen in relativ statischen Umgebungen, in denen sich zudem die beteiligten Entitäten untereinander kennen. Dies ist in mobilen Ad-hoc-Netzen, die auf der spontanen Vernetzung einander zuvor meist unbekannter mobiler Endgeräte beruhen, nicht der Fall. Somit sind auch subjektive Reputationssysteme schlecht in beliebigen mobilen Umgebungen einzusetzen.
- *Verteilte Lösungen* benötigen ähnlich den zentralisierten Lösungen den Zugriff auf eine Vertrauensinfrastruktur, die jedoch nicht zentral sondern peer-to-peer-artig organisiert ist. Im Extremfall eines infrastrukturlosen Single-Hop-Netzes fehlen jedoch die Kommunikationsmöglichkeiten, um die Verfügbarkeit dieser Infrastruktur sicherzustellen.

Somit scheint eine Lösung mit *lokaler Speicherung der Reputationsinformationen* am besten für die Nutzung in beliebigen mobilen Umgebungen geeignet zu sein, da sie lediglich die Möglichkeit einer direkten Kommunikation zwischen dem Bewertenden und dem Reputationsobjekt voraussetzt. Die Speicherung der Reputationsinformationen beim Reputationsobjekt wirft allerdings Fragen hinsichtlich ihrer Integrität auf.

### 6.2.1 Frage der Integrität von Bewertungen bei lokaler Speicherung

Zur korrekten Berechnung von Reputationswerten ist die Integrität der hierfür verwendeten Bewertungen unabdingbar. Diese kann an mehreren Stellen verletzt werden:

- *Bei der Abgabe der Bewertung:* Die Frage nach dem korrekten Verhalten des Bewertenden beim Abgeben seiner Bewertung entspricht einer weiteren Iteration der rekursiven Vertrauensfrage. Als Motivation für eine solche bewusste Falschbewertung wird in der Literatur zwischen der ungerechtfertigten Verbesserung eines Reputationswertes (engl. *Ballot Stuffing*) und seiner Verschlechterung (engl. *Bad Mouthing*) unterschieden.
- *Bei der Speicherung der Bewertung:* Hierbei ist sowohl die Unverfälschtheit einzelner Bewertungen als auch die Vollständigkeit aller gespeicherten Bewertungen sicherzustellen.

- *Bei der Aggregation der Reputation:* Dies läuft auf den Nachweis der totalen Korrektheit des Reputationsalgorithmus hinaus.
- *Bei der Verteilung der Reputation:* Dem kann mit den herkömmlichen Mechanismen zur Wahrung der Kommunikationssicherheit entgegen gewirkt werden.

Für Reputationssysteme mit lokaler Speicherung stellt die Sicherheit ebendieses Speichers eine besondere Herausforderung dar, da das Reputationsobjekt direkten Zugriff auf die über es abgegebenen Bewertungen hat. Die hierzu in der Literatur vorgeschlagenen Lösungen lassen sich folgenden Ansätzen zuordnen:

**Bestätigung durch eine vertrauenswürdige dritte Partei** In [FL02] wird die Interaktion zwischen zwei Peers eines P2P-Netzes durch eine zentrale Einheit, das sogenannte *Portal*, vermittelt. Zu Beginn einer Interaktion bestätigt dieses die Vollständigkeit der vom Reputationsobjekt übermittelten Bewertungsliste. Die Unverfälschtheit der einzelnen Bewertungen wird mittels digitaler Signaturen sichergestellt. Nach erfolgter Transaktion trägt das Portal dafür Sorge, dass sich beide Peers gegenseitig bewerten.

Einen ähnlichen Ansatz verfolgen Gupta et al. in [GJA03], die ebenfalls ein Reputationssystem für unstrukturierte P2P-Netze konstruieren. Auch diese Arbeit basiert auf einer vertrauenswürdigen dritten Partei, dem *Reputation Computation Agent (RCA)*. Dieser versieht die von einem Peer gespeicherte Reputation mit einer elektronischen Signatur und schützt sie so vor Manipulationen. Er muss daher in periodischen Abständen kontaktiert werden, um die Reputation aktuell zu halten.

In dem in [Vos04] beschriebenen Modell wird Reputation mittels elektronischer Münzen erfasst, um möglichst wenig Informationen über das Reputationsobjekt und den Bewertenden zu enthüllen. Je mehr Münzen ein Reputationsobjekt besitzt, desto zufriedener waren seine bisherigen Transaktionspartner mit ihm gewesen. Bei einer positiven Bewertung wird von einer vertrauenswürdigen dritten Partei eine entsprechende Menge personalisierter Münzen an das Reputationsobjekt ausgegeben. Durch die Personalisierung der Münzen können diese nur von dem Reputationsobjekt benutzt werden. Zu Beginn einer neuen Transaktion muss dieses eine vereinbarte Menge von Münzen als Pfand an seinen Transaktionspartner übergeben. Ist dieser mit dem Ausgang der Transaktion zufrieden, gibt er die Münzen zurück. Außerdem meldet er den positiven Ausgang der vertrauenswürdigen dritten Partei, die ihn hierfür mit einer Münze belohnt und auch an das Reputationsobjekt eine neue Münze ausgibt.

**Bestätigung durch andere Bewertende** Im Gegensatz zum zentralisierten Konzept einer vertrauenswürdigen dritten Instanz, verfolgt das in [LZBT03] vorgestellte System einen vollständig verteilten Ansatz. Die Reputationsdaten werden in sogenannten *RCerts*, einer signierten Liste der erhaltenen Bewertungen, gespeichert. Ein RCert besteht aus einem Kopf und einer beliebigen Anzahl von Bewertungen. Der Kopf enthält

allgemeine Informationen über das Reputationsobjekt, wie etwa seine Identität. Eine Bewertung enthält neben der Einschätzung des Reputationsobjekts die Identität des Bewertenden und den Zeitpunkt der Bewertung. Um die Integrität des RCert sicherzustellen, signiert der letzte Bewertende das gesamte RCert inklusive des Kopfes. Um darüber hinaus auch die Vollständigkeit des RCert zu garantieren, muss der aktuelle Bewertende zu Beginn einer geplanten Interaktion beim zuletzt aufgeführten Bewertenden den Zeitpunkt der von ihm abgegebenen Bewertung erfragen. Beim Abgeben der eigenen Bewertung muss er dann den letzten Bewertenden darüber informieren, dass eine neue Bewertung erfolgt ist und der zuvor übermittelte Zeitstempel damit seine Gültigkeit verloren hat.

**Verwendung von Trusted Hardware** Als dritte Alternative wird schließlich der Einsatz von Trusted Hardware in den beteiligten Endgeräten vorgeschlagen. Bei dem in in [VHM05] vorgestellten Ansatz berechnet sich der Reputationswert aus der Summe der ausschließlich positiven Bewertungen. Um die Integrität sicherzustellen, wird der Einsatz eines mit manipulationssicherer Hardware realisierten *Observers* [CP92] vorgeschlagen, der den Datenaustausch zwischen dem Bewertenden und dem Reputationsobjekt sowie die Speicherung der Reputationsinformationen kontrolliert.

Alle aufgeführten Ansätze erfüllen die Anforderungen des von ihnen adressierten Anwendungsszenarios. Sie lassen sich jedoch nicht ohne weiteres in mobilen Ad-hoc-Netzen einsetzen, in denen sich weder die permanente Erreichbarkeit der vertrauenswürdigen dritten Partei, noch die des letzten Bewertenden garantieren lässt. Auch die Verwendung manipulationssicherer Hardware ist nicht unproblematisch. Zwar enthalten immer mehr mobile Endgeräte ein *Trusted Platform Module (TPM)* gemäß den Spezifikationen der *Trusted Computing Group (TCG)*[3]. Es besteht aber ein Unterschied zwischen manipulationssicheren und vertrauenswürdigen Komponenten [PPSW97].

Im Folgenden werden daher zwei weitere Ansätze skizziert, um Manipulationen der gespeicherten Bewertungen durch das Reputationsobjekt zu unterbinden.

### 6.2.2 Fehlender Anreiz für die Manipulation von Bewertungen

Versucht ein Reputationsobjekt die von ihm gespeicherten Bewertungen zu verändern, dann geschieht dies mit dem Ziel, den eigenen Ruf zu verbessern. Es kann dafür entweder negative Bewertungen verfälschen bzw. ganz entfernen oder aber gefälschte positive Bewertungen hinzufügen. Der im Folgenden beschriebene Ansatz verwendet ausschließlich positive Bewertungen, um eine vollständige Speicherung sicherzustellen. Sonstige Manipulationen werden durch ein digitales Signatursystem unterbunden.

[3] http://www.trustedcomputinggroup.org/ (Abruf: September 2008)

**Lösungsansatz**

Zu Beginn wählt ein Teilnehmer $A$ ein asymmetrisches Schlüsselpaar $(sk_A, pk_A)$, dessen öffentlicher Teil $pk_A$ das Pseudonym darstellt, unter dem er im Reputationssystem agiert. Mit diesem muss er sich bei einer zentralen Zertifizierungsstelle (engl. *Certificate Authority – CA*) registrieren, die ihm darüber ein Zertifikat $Cert_{pk_A}$ ausstellt. Dabei erhält er auch den öffentlichen Schlüssel $pk_{CA}$ der CA, mit dem er die Zertifikate anderer Teilnehmer überprüfen kann.

Nach dem zufriedenstellenden Ausgang einer Interaktion kann er dann ein Reputationsobjekt bewerten. Hierzu konstruiert er eine Bewertung, die sich aus folgenden Bestandteilen zusammensetzt:

- Der Identität $id$ des bewerteten Reputationsobjekts.
- Der Repräsentation $z \in \mathbb{R}^+$ seiner Zufriedenheit über den Verlauf der Interaktion.
- Einem Zeitstempel $t$, um die mehrfache Verwendung einer gültigen Bewertung zu verhindern.
- Seinem öffentlichen Schlüssel $pk_A$ sowie des hierfür erteilten Zertifikats $Cert_{pk_A}$ zur Überprüfung seiner Authentizität.
- Einer mit seinem geheimen Schlüssel $sk_A$ erstellten Signatur $Sig := \text{sign}(sk_A, (id, z, t, pk_A, Cert_{pk_A}))$, um nachträgliche Manipulationen der Bewertung aufzudecken.

Das Reputationsobjekt fügt diese Bewertung seiner Bewertungsliste hinzu. Diese übermittelt es zu Beginn einer neuen Interaktion an den potentiellen Partner, der jede einzelne Bewertung anhand ihrer Signatur überprüft bevor er sie als Eingabe an den Reputationsalgorithmus übergibt.

**Diskussion**

Digitale Signatursysteme sind ein probates Mittel zur Gewährleistung der Integrität. Die nachträgliche Veränderung einer Bewertung kann damit ausgeschlossen werden. Gefälligkeitswertungen zwischen kollaborierenden Nutzern sind allerdings immer noch möglich.

Für die Verifikation der Authentizität des Signaturschlüssels sowie zur Verhinderung sogenannter Sybil-Angriffe [Dou02] benötigt der geschilderte Ansatz eine vertrauenswürdige Zertifizierungsstelle. Dies steht im Widerspruch zu dem bisher geforderten vollständigen Verzicht auf jegliche Infrastruktur. Es ist aber nur eine einmalige Registrierung und kein permanenter Zugriff auf die CA erforderlich, weshalb diese Ausnahme von der Regel akzeptabel erscheint.

Ein stärkerer Kritikpunkt ist die Beschränkung auf positive Bewertungen. Es bleibt unklar, ob dies für die Umsetzung stichhaltiger Reputationsmodelle ausreichend ist. So mag eine rein positive Bewertung zur Einschätzung der Kooperationsbereitschaft eines Reputationsobjekts noch genügen, bei den anderen semantischen Schutzzielen besteht jedoch größerer Klärungsbedarf. Insbesondere fällt die Unterscheidung zwischen schlechten und neuen Teilnehmern schwer. Unter Umständen lässt sich die Qualität des Reputationsalgorithmus zwar durch die Einbeziehung von Kontextinformationen, wie der Leistungskapazität des Reputationsobjekts, verbessern. Es besteht aber auch die Möglichkeit, dass beispielsweise die Einschätzung der Diskretion eines Reputationsobjekts nicht ohne negative Bewertungen auskommt. Im folgenden wird daher ein weiterer Lösungsansatz vorgestellt, der negative und positive Bewertungen erlaubt, diese aber so verschleiert, dass sie für das Reputationsobjekt nicht unterscheidbar sind.

### 6.2.3 Fehlendes Wissen für die Manipulation von Bewertungen

Der folgende Ansatz zur Verschleierung von Bewertungen ähnelt dem Prinzip der verdeckten impliziten Adressierung bei Broadcast-Anonymitätsmechanismen zum Schutz des Empfängers einer Nachricht [FP97, Seite 92]. Eine verdeckte implizite Adresse ist ein nur durch den Adressaten identifizierbares Merkmal einer Nachricht, mit dem er erkennen kann, dass sie für ihn bestimmt ist. Analog hierzu stellt das nun vorgestellte Verfahren sicher, dass die Bewertung eines Nutzers $A$ nur von seinem Bekanntenkreis interpretiert werden kann.

#### Lösungsansatz

Zu Beginn definiert ein Teilnehmer $A$ seine individuelle geordnete Bewertungsmenge $\mathcal{B}_A := \{b_{A1}, b_{A2}, \ldots, b_{An}\}$. Ihre Elemente repräsentieren die möglichen Abstufungen seiner Zufriedenheit mit dem Verlauf einer Interaktion. Um die tatsächliche Bedeutung einer Bewertung für das Reputationsobjekt und Außenstehende zu verschleiern, wird die Bewertungsmenge mit der Verschleierungsfunktion $v_A : \mathcal{B}_A \rightarrow \mathcal{S}_A$ eineindeutig auf eine gleichmächtige Menge asymmetrischer Schlüsselpaare $\mathcal{S}_A := \{(sk_{A1}, pk_{A1}), (sk_{A2}, pk_{A2}), \ldots, (sk_{An}, pk_{An})\}$ abgebildet. Seine Schlüsselmenge $\mathcal{S}_A$ lässt $A$ dann von einer zentralen Zertifizierungsstelle signieren und erhält so eine Zertifikatsliste $\mathcal{Z}_A := (Cert_{pk_{A1}}, Cert_{pk_{A2}}, \ldots, Cert_{pk_{An}})$ sowie den öffentlichen Zertifizierungsschlüssel $pk_{CA}$ der Zertifizierungsstelle. Die Bewertung einer Interaktion verläuft dann wie folgt:

- $A$ wählt gemäß seiner Zufriedenheit über den Verlauf der Interaktion einen Wert $b \in \mathcal{B}_A$ aus der Bewertungsmenge.
- Auf diesen wendet er die Verschleierungsfunktion $v_A(b)$ an und erhält so das korrespondierende asymmetrische Schlüsselpaar $(sk_b, pk_b)$.

- Nun konstruiert er seine Bewertung, die sich aus folgenden Bestandteilen zusammensetzt:
  - Der Identität $id$ des bewerteten Reputationsobjekts.
  - Einem Zeitstempel $t$, um die Wiederverwendung einer Bewertung zu verhindern.
  - Dem zuvor ermittelten öffentlichen Schlüssel $pk_b$ und des hierfür erstellten Zertifikats $Cert_{pk_b}$.
  - Einer mit dem zugehörigen geheimen Schlüssel $sk_b$ generierten Signatur $Sig := \text{sign}(sk_b, (id, t, pk_b, Cert_{pk_b}))$.

Das Reputationsobjekt fügt die Bewertung seiner lokalen Bewertungsliste hinzu, die es auf Anfrage einem neuen Interaktionspartner übermittelt. Genau wie das Reputationsobjekt ist dieser jedoch nicht in der Lage, den mit einer Bewertung ausgedrückten Grad an Zufriedenheit zu interpretieren. Hierfür benötigt er die Verschleierungsfunktion $v_A$, die ihm $A$ zur Verfügung stellen muss. Bildet er nun noch den Bewertungsmaßstab von $A$ auf seinen eigenen ab, kann er die Bewertungen von $A$ als Eingabe für seinen Reputationsalgorithmus verwenden.

#### Diskussion

Wie bei dem in Kapitel 6.2.2 geschilderten Ansatz erfolgt auch hier der Schutz der Bewertungen vor nachträglichen Veränderungen durch eine digitale Signatur. Durch die Verschleierung des Inhalts einer Bewertung wird außerdem die Vollständigkeit der Bewertungsliste sichergestellt, da das Reputationsobjekt nicht mehr in der Lage ist, ungünstige Wertungen von günstigen zu unterscheiden. Dadurch ermöglicht dieser Ansatz auch negative Bewertungen, womit ein Kritikpunkt der zuvor vorgeschlagenen Lösung entfällt.

Diese Eigenschaft hat allerdings ihren Preis. So kann die Interpretation einer Bewertung nur noch im „sozialen Kontext“ des Bewertenden erfolgen. Dieser muss seinem Umfeld die hierfür benötigten Informationen mittels geeigneter Protokolle und Mechanismen zur Verfügung stellen. Die Verbreitung der Verschleierungsfunktion ist dabei eine weitere Iteration der rekursiven Vertrauensfrage. Außerdem ist zu klären, ob solche sozialen Netze in dynamischen mobilen Umgebungen überhaupt in ausreichendem Maße existieren und wie stabil sie sind.

## 6.3 Zusammenfassung

Das vorliegende Kapitel gibt einen allgemeinen Überblick über die Eigenschaften von Reputationssystemen und ihre Anwendung zur Einschätzung der Verlässlichkeit in mobilen Umgebungen. Damit vermittelt es einen ersten Eindruck von der Komplexität,

aber auch der Tragweite dieser Thematik, ist die fehlende Vertrauensbasis doch ein essentielles Problem aller selbstorganisierenden Netze. Trotz zahlreicher Arbeiten in diesem Bereich sind noch viele Fragen offen, die auch hier nicht beantwortet werden konnten. Für eine ausführliche Aufzählung offener Forschungsfragen sei auf [JIB07, BMB08] verwiesen.

Ein grundlegendes Problem ist das Fehlen eines durch die Humanwissenschaften begründeten klaren Verständnisses der Phänomene Vertrauen und Reputation und damit das Fehlen eines mathematischen Modells hierfür. Als Folge hiervon verwenden die existierenden Reputationssysteme unterschiedlichste Reputationsmodelle, deren Stichhaltigkeit und Qualität in der Regel nicht nachgewiesen wird. Neben einem Einverständnis über die hierbei anzuwendenden Bewertungskriterien werden deshalb geeignete Simulations- und Testumgebungen benötigt.

Desweiteren ist die Diskussion der Sicherheit von Reputationssystemen zur Zeit noch unzureichend. Zwar wurde die Problematik grundsätzlich erkannt [CH07], es fehlt aber an einer systematischen Herangehensweise. So wäre eine genauere Analyse der in anderen Disziplinen beobachteten Angriffsszenarien wünschenswert, um Sicherheitsanforderungen für die einzelnen Komponenten und Nutzer eines Reputationssystems abzuleiten. Im Sinne mehrseitiger Sicherheit wären hierbei auch die Auswirkungen gegenläufiger Anforderungen zu untersuchen. Die Arbeit von Steinbrecher [Ste08] macht einen ersten Schritt in diese Richtung. Ihr Systementwurf betrachtet aber lediglich zentrale Reputationssysteme.

Ein weiterer Kritikpunkt bei den bisherigen Ansätzen zur Verbesserung der Sicherheit von Reputationssystemen ist ihre Beschränkung auf präventive Mechanismen. Vertrauensbasierte Entscheidungen enthalten aber ganz bewusst die Möglichkeit einer Fehleinschätzung, tolerieren also bis zu einem gewissen Grad einen Angriff. Es stellt sich daher die Frage, ob wirklich alle Sicherheitsanforderungen in Reputationssystemen präventiv durchgesetzt werden können oder ob es nicht sinnvoller ist, auch reaktive Mechanismen zu verwenden. Es erscheint deshalb lohnenswert, die Anwendbarkeit bekannter IDS-Ansätze auf Reputationssysteme zu untersuchen.

# 7
# Schlusswort und Ausblick

Sinkende Preise und steigende Leistungsfähigkeit haben in den letzten Jahren zu einer deutlich wachsenden Verbreitung mobiler Endgeräte geführt. Damit einher geht die Entwicklung innovativer Anwendungen und Dienste für mobile Netze. In der Folge werden immer mehr Daten auf mobilen Endgeräten gespeichert und über drahtlose Netze übertragen. Dies führt zu neuen Herausforderungen bei der Wahrung der Datensicherheit, da herkömmliche Mechanismen und Verfahren nur eingeschränkt oder überhaupt nicht für den Schutz mobiler Netze und Umgebungen geeignet sind.

Seit Ende der 1990er Jahre wurde eine Vielzahl von Versuchen unternommen, diesem Problem zu begegnen. Die Lösungsansätze aus der Industrie konzentrieren sich dabei auf infrastrukturelle Netzarchitekturen auf der einen und präventive Sicherheitsmechanismen auf der anderen Seite. Darüber hinausgehende Konzepte, etwa zum Schutz mobiler Ad-hoc-Netze, fehlen. Diese Fragestellung wird zwar in der Wissenschaftsgemeinde vielfach diskutiert, meist jedoch ohne in der Praxis evaluierte Ergebnisse zu liefern. Darüber hinaus betrachten quasi alle bisher vorgeschlagenen Lösungen nur einzelne Aspekte des komplexen Problems Sicherheit. Was fehlt ist ein praxistauglicher, ganzheitlicher Ansatz.

Das Ziel dieser Dissertation war daher die Entwicklung einer universellen Systemumgebung zur Integration existierender Sicherheitsmechanismen für mobile Ad-hoc- und Infrastrukturnetze. Die Arbeit kann hierzu folgende Beiträge leisten:

- Die Formulierung allgemeiner Forderungen für die Konzeption von Sicherheitsmechanismen für mobile Umgebungen, die mit deren Eigenschaften begründet werden, sowie hieraus abgeleiteter Anforderungen an eine entsprechende Sicherheitslösung.

- Ein strukturierter Vergleich existierender Sicherheitslösungen für IEEE 802.11-basierte Ad-hoc- und Infrastrukturnetze.
- Die Konzeption der hybriden universellen Sicherheitsarchitektur HUSAR zur Integration von Sicherheitsmechanismen in einem kooperativen Verbund mobiler Endgeräte.
- Die Validierung dieser Sicherheitsarchitektur durch die prototypische Entwicklung einer kooperativen Systemlösung zur praktischen Erprobung von verteilten Angriffserkennungsstrategien in Wireless LANs.
- Eine mittels semantischer Sicherheitsanforderungen formalisierte Darstellung der Vertrauensbildung in selbstorganisierenden mobilen Netzen und die Diskussion der Anwendung eines Reputationssystems zur Unterstützung ihrer Durchsetzung.

Das in dieser Arbeit entwickelte Intrusion Detection System ermöglicht grundsätzlich die anomalie- als auch die missbrauchsbasierte Erkennung von Angriffen, wenn auch zur Zeit ausschließlich der letztgenannte Ansatz verfolgt wird. Als weitere Einschränkung unterstützt der Prototyp in seiner jetzigen Form nur infrastrukturelles Wireless LAN. Es wurde aber eine allgemeine Methodik für die Unterstützung beliebiger Netzprotokolle beschrieben. Mit der sich abzeichnenden Verabschiedung des IEEE-Standards 802.11s und seiner Integration in aktuelle Betriebssysteme[1] ist eine Erweiterung für mobile Ad-hoc-Netze daher problemlos möglich.

Auch die Komposition lokaler Sicherheitsmechanismen für den verbesserten Schutz eines Endgeräts bleibt zukünftigen Arbeiten überlassen. Stattdessen konzentriert sich der entwickelte Prototyp auf die Organisation der Kooperation mehrerer mobiler Geräte. Fehlende Mechanismen zur effizienten Abstimmung von physischem und Overlay-Routing versucht die vorgestellte Implementierung durch die nachträgliche Berücksichtigung der räumlichen Position eines mobilen Knotens zu kompensieren. Als wichtige Grundlage für die Sicherheit in solch selbstorganisierenden Systemen ist ferner eine Schnittstelle zu einem Trust Management System vorgesehen.

Die Frage der Vertrauenswürdigkeit wird abschließend durch die Definition der Verlässlichkeit mobiler Knoten bezogen auf ihre Diskretion, Konformität und Kooperationsbereitschaft vertieft. Hierbei werden Reputationssysteme zur Bewertung der Verlässlichkeit vorgestellt und erste Lösungsansätze sowie offene Forschungsfragen für ihren Einsatz in mobilen Umgebungen diskutiert.

Raum für weitere Untersuchungen bieten schließlich die in dieser Arbeit nicht berücksichtigten Aspekte des Datenschutzes und der Gebrauchstauglichkeit. Letztere wurde zwar eingangs gefordert, die Implikationen der Gestaltung von Nutzeroberflächen für mobile Endgeräte im allgemeinen und solcher für mobile Sicherheitsmechanismen im besonderen sind jedoch zu weitreichend, um hier abgehandelt zu werden.

[1] Die im Sommer 2008 veröffentlichte Version 2.6.26 des Linux-Kernels unterstützt bereits den zu diesem Zeitpunkt gültigen Entwurf von IEEE 802.11s.

# A

# Dokumentation des Programmcodes

Dieser Anhang beschreibt den Inhalt der Web-Seite http://wwwrn.inf.tu-dresden.de/~gross/phd/, auf der der Quellcode der im Rahmen dieser Arbeit erstellten Prototypimplementierung bereitgestellt wird.

| Dateiname | Erläuterung |
|---|---|
| WIDS.zip | Quellcode des in Kapitel 4 konzipierten Wireless IDS |
| KooperationsFW.zip | Quellcode des in Kapitel 5 konzipierten Kooperations-Framework |
| License.html | Lizenzvereinbarung für die Nutzung des bereitgestellten Quellcodes |

# Abkürzungsverzeichnis

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AP | Access Point |
| ARAN | Authenticated Routing for Ad hoc Networks |
| AST | Abstract Syntax Tree |
| BSSID | Basic Service Set Identification |
| CA | Certificate Authority |
| CBC | Cipher Block Chaining |
| CBC-MAC | Cipher Block Chaining Message Authentication Code |
| CCM | Counter with CBC MAC |
| CCMP | Counter-Mode/CBC-MAC Protocol |
| CONFIDANT | Cooperation of Nodes: Fairness in Dynamic Ad hoc Networks |
| CORE | Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks |
| CRC | Cyclic Redundancy Check |
| CSM | Cooperating Security Managers |
| CTS | Clear To Send |
| DA | Destination Address |
| DHCP | Dynamic Host Configuration Protocol |
| DHT | Distributed Hash Table |
| DIDS | Distributed Intrusion Detection System |

| | |
|---|---|
| DoS | Denial-of-Service |
| DS | Distribution System |
| DSR | Dynamic Source Routing |
| DWSA | Distributed Wireless Security Auditor |
| EAP | Extensible Authentication Protocol |
| ESS | Extended Service Set |
| FCS | Frame Check Sequence |
| FIFO | First In - First Out |
| FMS | Fluhrer-Mantin-Shamir |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| HIDS | Host Intrusion Detection System |
| HMAC | Message Authentication Code |
| HTTP | Hypertext Transfer Protocol |
| HUSAR | Hybride universelle Sicherheitsarchitektur |
| ICMP | Internet Control Message Protocol |
| ID | Identifier |
| IDES | Intrusion Detection Expert System |
| IDMEF | Intrusion Detection Message Exchange Format |
| IDS | Intrusion Detection System |
| IEEE | Institute of Electrical and Electronics Engineers |
| I/O | Input/Output |
| IP | Internet Protocol |
| IPC | Inter-Process Communication |
| ISO | International Organization for Standardization |

| | |
|---|---|
| IV | Initialisierungsvektor |
| LAN | Local Area Network |
| LIDS | Local Intrusion Detection System |
| MAC | Media Access Control |
| MAN | Metropolitan Area Network |
| MANET | Mobiles Ad-hoc-Netz |
| MIC | Message Integrity Code |
| MitM | Man-in-the-Middle |
| MobIDS | Mobile Intrusion Detection System |
| NIDS | Network Intrusion Detection System |
| NSM | Network System Monitor |
| OSI | Open Systems Interconnection |
| OSRP | On-demand Secure Routing Protocol |
| OUI | Organizationally Unique Identifier |
| P2P | Peer-to-Peer |
| PDA | Personal Digital Assistant |
| PSK | Pre-Shared Key |
| RC4 | Ron's Code 4 |
| RFID | Radio Frequency Identification |
| RSN | Robust Security Network |
| RTS | Ready To Send |
| SA | Source Address |
| SAM | Sicherheitsarchitektur für Mobile Ad hoc Netzwerke |
| SAODV | Secure Ad hoc On-Demand Distance Vector Protocol |
| SDSR | Secure Dynamic Source Routing Protocol |

| | |
|---|---|
| SEAD | Secure Efficient Distance Vector Routing Protocol |
| SLSP | Secure Link State Routing Protocol |
| SMB | Server Message Block Protocol |
| SNMP | Simple Network Management Protocol |
| SRP | Secure Routing Protocol |
| SSID | Service Set Identifier |
| SSL | Secure Sockets Layer |
| SUCV | Statistically Unique and Cryptographically Verifiable |
| TCG | Trusted Computing Group |
| TCP | Transmission Control Protocol |
| TKIP | Temporal Key Integrity Protocol |
| TLS | Transport Layer Security |
| TPM | Trusted Platform Module |
| TSN | Transitional Security Network |
| TTP | Trusted Third Party |
| UDP | User Datagram Protocol |
| UI | User Interface |
| WAN | Wide Area Network |
| WEP | Wired Equivalent Privacy |
| WIDS | Wireless Intrusion Detection System |
| WLAN | Wireless Local Area Network |
| WPA | Wi-Fi Protected Access |
| WPAN | Wireless Personal Area Network |
| XML | Extensible Markup Language |
| ZBIDS | Zone-based Intrusion Detection System |

# Literaturverzeichnis


[ACF+00] ALLEN, Julia ; CHRISTIE, Alan ; FITHEN, William ; MCHUGH, John ; PICKEL, Jed ; STONER, Ed: State of the Practice of Intrusion Detection / Carnegie Mellon University. Version: Januar 2000. http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html. Pittsburgh, PA, USA, Januar 2000 (CMU/SEI-99-TR-028). – Technical Report

[ACL06] AIME, Marco D. ; CALANDRIELLO, Giorgio ; LIOY, Antonio: A wireless distributed intrusion detection system and a new attack mode. In: *Proceedings of the 11th IEEE Int. Symposium on Computers and Communications (ISCC 2006)*. Cagliari, Sardinia, Italy, Juni 2006, S. 35–40

[ACP+02] ALBERS, Patrick ; CAMP, Olivier ; PERCHER, Jean-Marc ; JOUGA, Bernard ; MÉ, Ludovic ; PUTTINI, Ricardo: Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches. In: *The 1st International Workshop on Wireless Information Systems (WIS-2002)*, 2002

[AG07] ARTZ, Donovan ; GIL, Yolanda: A survey of trust in computer science and the Semantic Web. In: *Journal of Web Semantics* 5 (2007), Juni, Nr. 2, S. 58–71. http://dx.doi.org/10.1016/j.websem.2007.03.002. – DOI 10.1016/j.websem.2007.03.002

[AHNRR02] AWERBUCH, Baruch ; HOLMER, David ; NITA-ROTARU, Cristina ; RUBENS, Herbert: An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In: *Proceedings of the 1st ACM Workshop on Wireless Security*. Atlanta, GA, USA, 2002, S. 21–30

[Aira] *Airjack toolkit for raw WLAN frame injection and reception*. http://sourceforge.net/projects/airjack/, Abruf: Mai 2008. – Projekt Web-Seite

[Airb] *AirPWN – A framework for 802.11 (wireless) packet injection*. http://airpwn.sourceforge.net/, Abruf: Mai 2008. – Projekt Web-Seite

[Airc] *Airsnarf – A rogue AP setup utility*. http://airsnarf.shmoo.com/, Abruf: Mai 2008. – Projekt Web-Seite

[Aird] *AirSnort Homepage*. http://airsnort.shmoo.com/, Abruf: Mai 2008. – Projekt Web-Seite

[Aire] *Air WLAN-Tools*. http://www.aircrack-ng.org/, Abruf: Mai 2008. – Projekt Web-Seite

[AMCB04] ANDRADE, Nazareno ; MOWBRAY, Miranda ; CIRNE, Walfredo ; BRASILEIRO, Francisco: When Can an Autonomous Reputation Scheme Discourage Free-riding in a Peer-to-Peer System? In: *Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid (CCGRID'04)*, 2004

[Amo99a] AMOROSO, Edward G.: *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*. Intrusion.Net Books, 1999

[Amo99b] *Kapitel* 6. Intrusion Correlation. In: [Amo99a], S. 145–167

[Amo99c] *Kapitel* 1. Introduction to Intrusion Detection. In: [Amo99a], S. 15–36

[And80] ANDERSON, James P.: *Computer Security Threat Monitoring and Surveillance*. 1980. – James P. Anderson Co, Fort Washington, PA

[And01] ANDERSON, Kelsey: Analysis of the Traffic on the Gnutella Network / University of California. San Diego, CA, USA, März 2001 (CSE222). – Forschungsbericht

[AO05] ARGYROUDIS, Patroklos G. ; O'MAHONEY, Donal: Secure Routing for Mobile Ad hoc Networks. In: *IEEE Communications Surveys & Tutorials* 7 (2005), Nr. 3, S. 2–21

[AR05] ABDUL-RAHMAN, Alfarez: *A Framework for Decentralised Trust Reasoning*, University of London, Diss., 2005. http://redkeydigital.com/papers/a_abdulrahman_thesis.pdf

[Arb01] ARBAUGH, William A.: *An Inductive Chosen Plaintext Attack against WEP/WEP2*. http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm. Version: Mai 2001. – Presentations to IEEE 802.11 TGi

[AW07] ANANTVALEE, Tiranuch ; WU, Jie: A Survey on Intrusion Detection in Mobile Ad Hoc Networks. In: XIAO, Yang (Hrsg.) ; DU, Ding-Zhu (Hrsg.) ; SHEN, Xuemin (Hrsg.): *Wireless / Mobile Network Security*. Springer Verlag, 2007, Kapitel 7, S. 170–196

[AWD04] ABOLHASAN, Mehran ; WYSOCKI, Tadeusz ; DUTKIEWICZ, Eryk: A review of routing protocols for mobile ad hoc networks. In: *Ad Hoc Networks* 2 (2004), Januar, Nr. 1, S. 1–22. http://dx.doi.org/10.1016/S1570-8705(03)00043-X. – DOI 10.1016/S1570–8705(03)00043–X

[Axe00] AXELSSON, Stefan: *Intrusion Detection Systems: A Survey and Taxonomy*. März 2000

[AY06] ANDEL, Todd R. ; YASINAC, Alec: On the credibility of manet simulations. In: *IEEE Computer* 39 (2006), Juli, Nr. 7, S. 48–54. http://dx.doi.org/10.1109/MC.2006.242. – DOI 10.1109/MC.2006.242

[Bac00a] BACE, Rebecca G.: *Intrusion Detection*. Macmillan Technical Publishing, 2000

[Bac00b] *Kapitel* 5. Responses. In: [Bac00a], S. 121–134

[Bac00c] *Kapitel* 3. Information Sources. In: [Bac00a], S. 45–78

[BB02] BUCHEGGER, Sonja ; BOUDEC, Jean-Yves L.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness in Dynamic Ad-hoc NeTworks). In: *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*. Lausanne, CH, 2002, S. 226–236

[BCC06] BEYAH, Raheem A. ; CORBETT, Cherita L. ; COPELAND, John A.: The Case for Collaborative Distributed Wireless Intrusion Detection Systems. In: *Proceedings of the IEEE International Conference on Granular Computing (GrC)*, 2006

[BEGA03] BOBBA, Rakesh B. ; ESCHENAUER, Laurent ; GLIGOR, Virgil ; ARBAUGH, William: Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. In: *IEEE Global Telecommunications Conference 2003 (GLOBECOM '03)* Bd. 3, 2003, S. 1511–1515

[BGW00] BORISOV, Nikita ; GOLDBERG, Ian ; WAGNER, David: *Security of the WEP algorithm*. Version: 2000. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html, Abruf: Mai 2008. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

[BGW01] BORISOV, Nikita ; GOLDBERG, Ian ; WAGNER, David: Intercepting Mobile Communications: The Insecurity of 802.11. In: *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. Rome, Italy, Juli 2001, S. 180–189

[BH00] BUTTYÁN, Levente ; HUBAUX, Jean-Pierre: Enforcing Service Availability in Mobile Ad-Hoc WANs. In: *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*. Bostan, MA, USA, 2000, 87–96

[BH03] BUTTYÁN, Levente ; HUBAUX, Jean-Pierre: Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. In: *Mobile Networks and Applications* 8 (2003), Oktober, Nr. 5, S. 579–592. http://dx.doi.org/10.1023/A:1025146013151. – DOI 10.1023/A:1025146013151

[BHL06] BITTAU, A. ; HANDLEY, M. ; LACKEY, J.: The final nail in WEP's coffin. In: *Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*, 2006

[Bis03] BISHOP, Matt: *Computer security: art and science*. Pearson Education, Inc., 2003

[BMB08] BUCHEGGER, Sonja ; MUNDINGER, Jochen ; BOUDEC, Jean-Yves L.: Reputation Systems for Self-Organized Networks. In: *IEEE Technology and Society Magazine* 27 (2008), Nr. 1, S. 41–47. http://dx.doi.org/10.1109/MTS.2008.918039. – DOI 10.1109/MTS.2008.918039. – Special Issue on Limits of Cooperation in Wireless Communications

[BNPDS04] BRANCH, Joel W. ; NICK PETRONI, Jr. ; DOORN, Leendert V. ; SAFFORD, David: Autonomic 802.11 Wireless LAN Security Auditing. In: *IEEE Security and Privacy Magazine* 2 (2004), Mai, Nr. 3, S. 56–65

[BS03] BELLARDO, John ; SAVAGE, Stefan: 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In: *Proceedings of the 12th USENIX Security Symposium*. Washington D.C., USA, August 2003, 15–28

[Bul06a] BULK, Frank: Safe Inside a Bubble. In: *Network Computing* 6.22 (2006), S. 26–40

[Bul06b] BULK, Frank: WIDPS Overlays Provide Ultra Tight Security. In: *Network Computing* 6.22 (2006), S. 1–12

[Cam03] CAMP, L. J.: Designing for Trust. Version: 2003. http://www.springerlink.com/content/uly0jda2dy75db71. In: FALCONE, Rino (Hrsg.) ; BARBER, Suzanne (Hrsg.) ; KORBA, Larry (Hrsg.) ; SINGH, Munindar (Hrsg.): *Trust, Reputation, and Security: Theories and Practice* Bd. 2631/2003. Springer Berlin/Heidelberg, 2003, 15–29

[Can01] CANAVAN, John E.: *Fundamentals of Network Security*. Artech House, 2001 http://acm.books24x7.com/viewer_r.asp?bookid=3491

[Cap04] CAPRA, Licia: Engineering Human Trust in Mobile System Collaborations. In: *Proceedings of the 12th International Symposium on the Foundations of Software Engineering (SIGSOFT 2004/FSE-12)*. Newport Beach, CA, USA, November 2004, 107–116

[CDV+02] CORNELLI, Fabrizio ; DAMIANI, Ernesto ; VIMERCATI, Sabrina De C. ; PARABOSCHI, Stefano ; SAMARATI, Pierangela: Choosing Reputable Servents in a P2P Network. In: *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*. Honolulu, Hawai, USA, Mai 2002, S. 376–386

[CH07] CARRARA, Elisabetta (Hrsg.) ; HOGBEN, Giles (Hrsg.): *Reputation-based Systems: a security analysis*. European Network and Information Security Agency (ENISA), 2007 (ENISA Position Paper 2). http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_pp_reputation_based_system.pdf

[Cha06] CHANDRA, Ranveer: *A Virtualization Architecture for Wireless Network Cards*. Ithaca, NY, USA, Cornell University, Diss., 2006. http://portal.acm.org/citation.cfm?id=1144933&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618#

[CM99] CORSON, S. ; MACKER, J.: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations / IETF Network Working Group. Version: Januar 1999. ftp://ftp.rfc-editor.org/in-notes/rfc2501.txt. 1999 (2501). – RFC

[CP92] CHAUM, David ; PEDERSEN, Torben P.: Wallet Databases with Observers. In: *Proceedings of the 12th Annual International Cryptology Conference* Bd. 740/1993. Santa Barbara, CA, USA, August 1992 (LNCS), S. 89–105

[CPZ06] CHANDRA, Ranveer ; PADMANABHAN, Venkata N. ; ZHANG, Ming: WiFiProfiler: Cooperative Diagnosis in Wireless LANs. In: *MobiSys 2006: Proceedings of the 4th international conference on Mobile systems, applications and services*. New York, NY, USA : ACM Press, 2006. – ISBN 1–59593–195–3, S. 205–219

[CSS02] CAVIN, David ; SASSON, Yoav ; SCHIPER, André: On the Accuracy of MANET Simulators. In: *Proceedings of the 2nd ACM International Workshop Principles of Mobile Computing*. Toulouse, France, 2002, S. 38–43

[DCF06] DEBAR, H. ; CURRY, D. ; FEINSTEIN, B.: The Intrusion Detection Message Exchange Format / IETF Intrusion Detection Working Group. Version: Februar 2006. http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-16.txt. 2006 (draft-ietf-idwg-idmef-xml-15). – Internet-Draft

[Del03] DELLAROCAS, Chrysanthos: The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. In: *Management Science* 49 (2003), Oktober, Nr. 10, S. 1407–1424. http://dx.doi.org/10.1287/mnsc.49.10.1407.17308. – DOI 10.1287/mnsc.49.10.1407.17308

[Den87] DENNING, Dorothy E.: An Intrusion-Detection Model. In: *IEEE Transactions on Software Engineering* SE-13 (1987), Februar, Nr. 2, S. 222–232

[DGK+03] DRESCHER, Peter ; GERWING, Heinz ; KÜGLER, Dennis ; NIEDERMEYER, Frank ; PÜTZ, Wilhelm ; RECKHAUS, Guido ; RASTEN, Robert ; SCHULTE-GEERS, Ernst ; TERNES, Berthold ; WIEMERS, Andreas: Drahtlose lokale Kommunikationssysteme und ihre Sicherheitsaspekte / Bundesamt für Sicherheit in der Informationstechnik, „Projektgruppe Local Wireless Communication“. Version: 2003. http://www.bsi.de/. 2003. – Informationsbroschüre

[DKA05] DIVAC-KRNIC, Luka ; ACKERMANN, Ralf: Security-Related Issues in Peer-to-Peer Networks. Version: 2005. http://dx.doi.org/10.1007/11530657. In: [SW05], Kapitel 31, S. 529–545

[DKPS05] DREGER, Holger ; KREIBICH, Christian ; PAXSON, Vern ; SOMMER, Robin: Enhancing the Accuracy of Network-based Intrusion Detection with Host-based Context. In: *Proceedings of the 2nd Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA'05)*, 2005

[DM02] DEBAR, Hervé ; MORIN, Benjamin: Evaluation of the Diagnostic Capabilities of Commercial Intrusion Detection Systems. In: *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection* Bd. 2516/2002, 2002 (LNCS), S. 177–198

[Dou02] DOUCEUR, John R.: The Sybil Attack. In: *Peer-to-Peer Systems: Proceedings of the 1st International Workshop on Peer-To-Peer Systems (IPTPS 2002)* Bd. 2429/2002. Cambridge, MA, USA : Springer Berlin/Heidelberg, Februar 2002 (Lecture Notes in Computer Science), 251–260

[Dud07] *Duden – Deutsches Universalwörterbuch*. 6., überarbeitete Auflage. Mannheim, Leipzig, Wien, Zürich : Dudenverlag, 2007

[DXL+06] DENG, Hongmei ; XU, Roger ; LI, Jason ; ZHANG, Frank ; LEVY, Renato ; LEE, Wenke: Agent-based cooperative anomaly detection for wireless ad hoc networks. In: *12th International Conference on Parallel and Distributed Systems (ICPADS 2006)*, 2006

[Dö06] DÖRHÖFER, Stefan: *Empirische Untersuchungen zur WLAN-Sicherheit mittels Wardriving*, RWTH Aachen, Diplomarbeit, September 2006

[EA04a] *Kapitel* 15. Known Attacks: Technical Review. In: [EA04b], S. 311–336

[EA04b] EDNEY, Jon ; ARBAUGH, William A.: *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley, 2004

[Eck03] ECKERT, Claudia: *IT-Sicherheit*. 2. Auflage. Oldenbourg Wissenschaftsverlag GmbH, 2003

[Eng07] ENGLER, Michael: *Fundamental Models and Algorithms for a Distributed Reputation System*, Universität Stuttgart, Diss., Dezember 2007. http://elib.uni-stuttgart.de/opus/volltexte/2008/3401/pdf/dissertation.pdf

[ES05] EBERSPÄCHER, Jörg ; SCHOLLMEIER, Rüdiger: First and Second Generation of Peer-to-Peer Systems. Version: 2005. http://dx.doi.org/10.1007/11530657_5. In: [SW05], Kapitel 5, S. 35–56

[ESZK04] EBERSPÄCHER, Jörg ; SCHOLLMEIER, Rüdiger ; ZÖLS, Stefan ; KUNZMANN, Gerald: Structured P2P Networks in Mobile and Fixed Environments. In: *Proceedings of the International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs'04)*, 2004. – Tutorial

[Fak] *FakeAP beacon frame generator*. http://www.blackalchemy.to/project/fakeap/, Abruf: Mai 2008. – Projekt Web-Seite

[FC05] FELDMAN, Michael ; CHUANG, John: Overcoming Free-Riding Behavior in Peer-to-Peer Systems. In: *ACM SIGecom Exchanges* 5 (2005), Juli, Nr. 4, S. 41–50. http://dx.doi.org/10.1145/1120717.1120723. – DOI 10.1145/1120717.1120723

[Fed98] FEDERRATH, Hannes: *Vertrauenswürdiges Mobilitätsmanagement in Telekommunikationsnetzen*, Technisch Universität Dresden, Diss., Februar 1998

[FL02] FAHRENHOLTZ, Dietrich ; LAMERSDORF, Winfried: Transactional Security for a Distributed Reputation Management System. In: *Proceedings of the 3rd International Conference on Electronic Commerce and Web Technologies* Bd. 2455/2002, 2002 (LNCS), S. 214–223

[Fle01] FLEISCH, Brett D.: *Intrusion Detection*. Vorlesungsfolien, Course CS165 - Computer Security. http://www.cs.ucr.edu/~brett/cs165_s01/LECTURE23/intrusion-4up.pdf. Version: Mai 2001. – http://www.cs.ucr.edu/~brett/cs165_s01/LECTURE23/intrusion-4up.pdf (letzter Abruf: Juli 2008)

[FMS01] FLUHRER, Scott ; MANTIN, Itsik ; SHAMIR, Adi: Weaknesses in the Key Scheduling Algorithm of RC4. In: VAUDENY, S. (Hrsg.) ; YOUSSEF, A.

(Hrsg.): *Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography (SAC 2001)* Bd. 2259/2001, Springer, August 2001 (LNCS), S. 1–24

[FP97] FEDERRATH, Hannes ; PFITZMANN, Andreas: Bausteine zur Realisierung mehrseitiger Sicherheit. In: [MP97b], S. 83–104

[GC05] GUO, Fanglu ; CHIUEH, Tzi cker: Sequence Number-Based MAC Address Spoof Detection. In: *8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005)* Bd. 3858/2006. Seattle, WA, USA : Springer Verlag, September 2005 (LNCS), S. 309–329

[Gei03] GEIER, Jim: *Identifying Rogue Access Points*. Version: Januar 2003. http://www.wi-fiplanet.com/tutorials/article.php/1564431, Abruf: Mai 2008. Wi-Fi Planet Tutorials

[GGPS97] GATTUNG, Gunther ; GRIMM, Rüdiger ; PORDESCH, Ulrich ; SCHNEIDER, Michael J.: Persönliche Sicherheitsmanager in der virtuellen Welt. In: [MP97b], S. 181–205

[GHJV95] *Kapitel* 5. Behavioral Patterns. In: GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995, S. 221–349

[GJA03] GUPTA, Minaxi ; JUDGE, Paul ; AMMAR, Mostafa: A Reputation System for Peer-to-Peer Networks. In: *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'03)*, 2003, S. 144–152

[GLR05] GERLA, Mario ; LINDEMANN, Christoph ; ROWSTRON, Ant: P2P MANET's – New Research Issues. In: GERLA, Mario (Hrsg.) ; LINDEMANN, Christoph (Hrsg.) ; ROWSTRON, Antony (Hrsg.): *Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks – New Research Issues*. Dagstuhl, Germany : Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005 (Dagstuhl Seminar Proceedings 05152)

[GN01] GANGER, Gregory R. ; NAGLE, David F.: Better Security via Smarter Devices. In: *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems*, 2001, 100–105

[GN06] GROSS, Stephan ; NEUMERKEL, René: A Sophisticated Solution for Revealing Attacks on Wireless LAN. In: FISCHER-HÜBNER, Simone (Hrsg.) ; FURNELL, Steven (Hrsg.) ; LAMBRINOUDAKIS, Costas (Hrsg.): *Trust, Privacy, and Security in Digital Business. 3rd International Conference, TrustBus*

*2006* Bd. 4083. Krakau, Polen : Springer Verlag, September 2006 (LNCS), S. 223–232

[GNV+92] GILHAM, Fred ; NEUMANN, Peter ; VALDES, Alfonso ; LUNT, Teresa F. ; TAMARU, Ann ; JAGANNATHAN, R. ; JALALI, Caveh ; JAVITZ, Harold S. ; GARVEY, Thomas D.: A Real-Time Intrusion-Detection Expert System (IDES) / SRI International. 1992. – Final Technical Report

[Gro06a] GROSS, Stephan: Selbstschützende mobile Systeme. In: DITTMANN, Jana (Hrsg.) ; Gesellschaft für Informatik (Veranst.): *3. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik (Sicherheit 2006)* Bd. P-77. Magdeburg, Februar 2006 (GI-Edition – Lecture Notes in Informatics (LNI) P-77), S. 103–106

[Gro06b] GROSS, Stephan: Towards Cooperative Self-Protecting Mobile Devices using Trustful Relationships. In: *2nd IEEE SECURECOMM SECOVAL Workshop: The Value of Security through Collaboration*. Baltimore, MD, USA, September 2006

[HBS73] HEWITT, Carl ; BISHOP, Peter ; STEIGER, Richard: A Universal Modular ACTOR Formalism for Artificial Intelligence. In: *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*. Stanford, CA, USA, August 1973

[HDL+90] HEBERLEIN, L. T. ; DIAS, Gihan V. ; LEVITT, Karl N. ; MUKHERJEE, Biswanath ; WOOD, Jeff ; WOLBER, David: A Network Security Monitor. In: *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*. Oakland, CA, USA, Mai 1990, S. 296–304

[HJP02] HU, Yih-Chun ; JOHNSON, David B. ; PERRIG, Adrian: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In: *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*, 2002, S. 3–13

[HL03] HUANG, Y. ; LEE, Wenke: A Cooperative Intrusion Detection System for Ad Hoc Networks. In: *SASN '03: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. New York, NY, USA : ACM Press, 2003. – ISBN 1–58113–783–4, S. 135–147

[HP04] HU, Yih-Chun ; PERRIG, Adrian: A Survey of Secure Wireless Ad Hoc Routing. In: *IEEE Security & Privacy* 2 (2004), Mai, Nr. 3, S. 28–39. http://dx.doi.org/10.1109/MSP.2004.1. – DOI 10.1109/MSP.2004.1

[HPJ05] HU, Yi-Chun ; PERRIG, Adrian ; JOHNSON, David B.: Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In: *Wireless Networks* 11 (2005), Januar, Nr. 1–2, S. 21–38. http://dx.doi.org/10.1007/s11276-004-4744-y. – DOI 10.1007/s11276–004–4744–y

[HPT97] HAUSER, Ralf ; PRZYGIENDA, Tony ; TSUDIK, Gene: Reducing the Cost of Security in Link-State Routing. In: *Proceedings of the 1997 Symposium on Network and Distributed System Security*. San Diego, CA, USA, Februar 1997, S. 93–99

[ISO05] *Information technology – Security techniques – Information security management systems – Requirements*. International Organization for Standardization, 2005 (ISO/IEC Standard 27001)

[JIB07] JØSANG, Audun ; ISMAIL, Roslan ; BOYD, Colin: A Survey of Trust and Reputation Systems for Online Service Provision. In: *Decision Support Systems* 43 (2007), Nr. 2, S. 618–644. http://dx.doi.org/10.1016/j.dss.2005.05.019. – DOI 10.1016/j.dss.2005.05.019

[Jøs96] JØSANG, Audun: The right type of trust for distributed systems. In: MEADOWS, C. (Hrsg.): *Proceedings of the 1996 Workshop on New Security Paradigms*. Lake Arrowhead, CA, USA : ACM Press, 1996, S. 119–131

[Kar03] KARGL, Frank: *Sicherheit in Mobilen Ad hoc Netzwerken*, Universität Ulm, Diss., 2003

[Kis] *KISMET – Wireless Sniffing and Monitoring*. http://www.kismetwireless.net/, Abruf: Mai 2008. – Projekt Web-Seite

[Kle08] KLEIN, Andreas: Attacks on the RC4 stream cipher. In: *Designs, Codes and Cryptography* Online First (2008), April. http://dx.doi.org/10.1007/s10623-008-9206-6. – DOI 10.1007/s10623–008–9206–6

[KLR05] KHOPKAR, Tapan ; LI, Xin ; RESNICK, Paul: Self-Selection, Slipping, Salvaging, Slacking, and Stoning: the Impacts of Negative Feedback at eBay. In: *Proceedings of the 6th ACM Conference on Electronic Commerce (EC'05)*. Vancouver, BC, Canada, 2005, S. 223–231

[KNG+04] KOTZ, David ; NEWPORT, Calvin ; GRAY, Robert S. ; LIU, Jason ; YUAN, Yougu ; ELLIOTT, Chip: Experimental evaluation of wireless simulation assumptions. In: *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems.* Venice, Italy, 2004, S. 78–82

[Kor04a] KOREK: *chopchop (Experimental WEP attacks)*. Version: September 2004. http://www.netstumbler.org/showthread.php?t=12489, Abruf: Mai 2008. – Beitrag in Online-Forum

[Kor04b] KOREK: *Next generation of WEP attacks?* Version: September 2004. http://www.netstumbler.org/93942-post35.html, Abruf: Mai 2008. – Beitrag in Online-Forum

[KR07a] KUROSE, James F. ; ROSS, Keith W.: *Computer Networking: A Top-Down Approach*. 4th edition. Pearson Education, Inc., 2007

[KR07b] *Kapitel* 4.7 Broadcast and Multicast Routing. In: [KR07a], S. 402–415

[KS05] KREIBICH, Christian ; SOMMER, Robin: Policy-controlled Event Management for Distributed Intrusion Detection. In: *Proceedings of the 4th International Workshop on Distributed Event-Based Systems (DEBS'05)*. Columbus, Ohio, USA, 2005

[Lam81] LAMPORT, Leslie: Password Authentication with Insecure Communication. In: *Communications of the ACM* 24 (1981), November, Nr. 11, S. 770–772

[LAN99] LAN MAN STANDARDS COMMITTEE OF THE IEEE COMPUTER SOCIETY: *802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standard. http://standards.ieee.org/getieee802/download/802.11-1999.pdf. Version: Juni 1999

[Leh07] LEHMANN, Robert: *Klassifikation und Modellierung von Angriffen auf Wireless LAN*, Technische Universität Dresden, Diplomarbeit, Januar 2007

[LSP82] LAMPORT, Leslie ; SHOSTAK, Robert ; PEASE, Marshall: The Byzantine Generals Problem. In: *ACM Transactions on Programming Languages* 4 (1982), Juli, Nr. 3, 382–401. http://portal.acm.org/citation.cfm?id=357176

[LW06] LINDEMANN, Christoph ; WALDHORST, Oliver P.: Peer-to-Peer-Systeme für drahtlose Multihop-Netze. In: *Informatik Spektrum* 29 (2006), Juni, Nr. 3, S. 222–226. http://dx.doi.org/10.1007/s00287-006-0076-x. – DOI 10.1007/s00287–006–0076–x

[LZBT03] LIAU, Chu Y. ; ZHOU, Xuan ; BRESSAN, Stéphane ; TAN, Kian-Lee: Efficient Distributed Reputation Scheme for Peer-to-Peer Systems. In: *Proceedings of the 2nd International Conference on Human.Society@Internet (HSI'03)* Bd. 2713/2003, 2003 (LNCS), S. 54–63

[MA02] MISHRA, Arunesh ; ARBAUGH, William A.: An Initial Security Analysis of the IEEE 802.1X Standard / University of Maryland. 2002 (CS-TR-4328). – Forschungsbericht. – UMIACS-TR-2002-10

[Mar94] MARSH, Stephan P.: *Formalising Trust as a Computational Concept*, University of Stirling, Diss., 1994. http://www.cs.stir.ac.uk/research/publications/techreps/pdf/TR133.pdf

[McH01] MCHUGH, John: Intrusion and intrusion detection. In: *International Journal of Information Science* 1 (2001), August, Nr. 1, S. 14–35. http://dx.doi.org/10.1007/s102070100001. – DOI 10.1007/s102070100001

[MDM07] MERWE, Johann Van D. ; DAWOUD, Dawoud ; MCDONALD, Stehen: A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks. In: *ACM Computing Surveys (CSUR)* 39 (2007), April, Nr. 1, S. 1–45. http://dx.doi.org/10.1145/1216370.1216371. – DOI 10.1145/1216370.1216371

[MGLB00] MARTI, Sergio ; GIULI, T. J. ; LAI, Kevin ; BAKER, Mary: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, S. 255–265

[MGM06] MARTI, Sergio ; GARCIA-MOLINA, Hector: Taxonomy of trust: Categorizing P2P reputation systems. In: *Computer Networks* 50 (2006), Nr. 4, S. 472–484. http://dx.doi.org/10.1016/j.comnet.2005.07.011. – DOI 10.1016/j.comnet.2005.07.011. – ISSN 1389–1286

[MM02] MICHIARDI, Pietro ; MOLVA, Refik: CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks. In: *Proceedings of the 6th IFIP Communication and Multimedia Security Conference (CMS'02)*. Portoroz, Slovenia, September 2002

[MMH02] MUI, Lik ; MOHTASHEMI, Mojdeh ; HALBERSTADT, Ari: Notions of Reputation in Multi-Agents Systems: A Review. In: *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*. Bologna, Italy, Juli 2002, S. 280–287

[MP97a] MÜLLER, Günter ; PFITZMANN, Andreas: Mehrseitige sichere Kommunikation – Vertrauen in Technik durch Technik. [MP97b], S. 11–18

[MP97b] MÜLLER, Günter (Hrsg.) ; PFITZMANN, Andreas (Hrsg.): *Mehrseitige Sicherheit in der Kommunikationstechnik*. Addison-Wesley, 1997

[Mui03] MUI, Lik: *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*, Massachusetts Institute of Technology, Diss., 2003

[Net] *NetStumbler Wireless Networking Tool*. http://www.netstumbler.com/, Abruf: Mai 2008. – Projekt Web-Seite

[Neu05] NEUMERKEL, René: *Entwicklung eines Angriffssensors für Wireless LANs*, Technische Universität Dresden, Diplomarbeit, Oktober 2005

[Oxf89] *Oxford Advanced Learner's Dictionary of Current English*. 4th edition. Oxford University Press, 1989

[Pax99] PAXSON, Vern: Bro: A System for Detecting Network Intruders in Real-Time. In: *Computer Networks* 31 (1999), Dezember, Nr. 23-24, S. 2435–2463. http://dx.doi.org/10.1016/S1389-1286(99)00112-7. – DOI 10.1016/S1389–1286(99)00112–7

[Per01] PERKINS, Charles E. (Hrsg.): *Ad hoc networking*. Addison-Wesley, 2001

[Pfi00] PFITZMANN, Andreas: *Sicherheit in Rechnernetzen: Mehrseitige Sicherheit in verteilten und durch verteilte Systeme*. Vorlesungsskript, 2000

[Pfi06] PFITZMANN, Andreas: Multilateral Security: Enabling Technologies and Their Evaluation. In: MÜLLER, Günter (Hrsg.): *Emerging Trends in Information and Communications Security (ETRICS 2006)* Bd. 3995/2006, 2006 (LNCS), S. 1–13

[PH02] PAPADIMITRATOS, Panagiotis ; HAAS, Zygmunt J.: Secure Routing for Mobile Ad hoc Networks. In: *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002

[PH03] PAPADIMITRATOS, Panagiotis ; HAAS, Zygmunt J.: Secure Link State Routing for Mobile Ad Hoc Networks. In: *Proceedings of the 2003 Symposium on Applications and the Internet Workshops*, 2003, 379–383

[Plu82] PLUMMER, David C.: An Ethernet Address Resolution Protocol / IETF Network Working Group. Version: November 1982. ftp://ftp.rfc-editor.org/in-notes/rfc826.txt. 1982 (826). – RFC

[PM07] PAZAND, Babk ; MCDONALD, Chris: A Critique of Mobility Models for Wireless Network Simulation. In: *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*. Melbourne, Australia, Juli 2007, S. 141–146

[PMPS04] PUTTINI, Ricardo ; MÉ, Ludovic ; PERCHER, Jean-Marc ; SOUSA, Rafael de: A Fully Distributed IDS for MANET. In: *Proceedings of the 9th IEEE Symposium on Computers and Communications (ISCC'2004)*, 2004

[PN97] PORRAS, P. A. ; NEUMANN, P. G.: EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: *National Information Systems Security Conference*. Baltimore, MD, USA, Oktober 1997

[PPM+03] PUTTINI, R. S. ; PERCHER, J.-M. ; MÉ, L. ; CAMP, O. ; SOUSA JR., R. de ; ABBAS, C. J. B. ; VILLALBA, L. J. G.: A Modular Architecture for Distributed IDS in MANET. In: *Proceedings of the 2003 International Conference on Computational Science and Its Applications (ICCSA)* Bd. 2669/2003, 2003 (LNCS)

[PPSW97] PFITZMANN, Andreas ; PFITZMANN, Birgit ; SCHUNTER, Matthias ; WAIDNER, Michael: Trusting Mobile User Devices and Security Modules. In: *IEEE Computer* 30 (1997), Februar, Nr. 2, S. 61–68. http://dx.doi.org/10.1109/2.566159. – DOI 10.1109/2.566159

[PZC+96] PUKETZA, Nicholas J. ; ZHANG, Kui ; CHUNG, Mandy ; MUKHERJEE, Biswanath ; OLSSON, Ronald A.: A Methodology for Testing Intrusion Detection Systems. In: *IEEE Transactions on Software Engineering* 22 (1996), Oktober, Nr. 10, S. 719–729. http://dx.doi.org/10.1109/32.544350. – DOI 10.1109/32.544350

[RBP+93] RUMBAUGH, James ; BLAHA, Michael ; PREMERLANI, William ; EDDY, Frederick ; LORENSEN, William: *Objektorientiertes Modellieren und Entwerfen*. Carl Hanser Verlag, 1993

[RD01] ROWSTRON, Antony I. T. ; DRUSCHEL, Peter: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)* Bd. 2218. Heidelberg, Germany, 2001 (Lecture Notes In Computer Science), S. 329–350

[Rec04] *Kapitel* 4.5 802.11-Frameformat. In: RECH, Jörg: *Wireless LANs*. 1. Auflage. Heise Zeitschriften Verlag GmbH & Co. KG, 2004, S. 171–204

[Ree79] REENSKAUG, Trygve: Thing-Model-View-Editor: an Example from a Planning System / Xerox PARC. Version: Mai 1979. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html. 1979. – Technical Note

[Rei06] REICHERT, Sandro: *Entwicklung eines verteilten Intrusion Detection Systems für Wireless LAN*, Technische Universität Dresden, Großer Beleg, Oktober 2006

[Rei07] REICHERT, Sandro: *Entwicklung einer Management-Komponente für ein verteiltes IDS auf mobilen Endgeräten*, Technische Universität Dresden, Diplomarbeit, Oktober 2007

[RIF02] RIPEANU, Matei ; IAMNITCHI, Adriana ; FOSTER, Ian: Mapping the Gnutella Network. In: *IEEE Internet Computing* 6 (2002), Nr. 1, S. 50–57. http://dx.doi.org/10.1109/4236.978369. – DOI 10.1109/4236.978369

[Rip01] RIPEANU, Matei: Peer-to-Peer Architecture Case Study: Gnutella Network / University of Chicago, Department of Computer Science. Version: Juli 2001. http://www.cs.uchicago.edu/research/publications/techreports/TR-2001-26. 2001 (TR-2001-26). – Forschungsbericht

[Rit01] RITTER, Jordan: *Why Gnutella Can't Scale. No, Real.* http://www.darkridge.com/~jpr5/doc/gnutella.html. http://www.darkridge.com/~jpr5/doc/gnutella.html. Version: Februar 2001. – letzter Zugriff: Juli 2008

[RJ96] RASMUSSON, Lars ; JANSSON, Sverker: Simulated Social Control for Secure Internet Commerce. In: *Proceedings of the 1996 ACM New Security Paradigm Workshop (NSPW'96)*. Lake Arrowhead, California, USA, 1996, 18–25

[RKK07] RUOHOMAA, Sini ; KUTVONEN, Lea ; KOUTROULI, Eleni: Reputation Management Survey. In: *Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES'07)*, 2007, S. 103–111

[Rot05a] *Kapitel* 11. Mobile Endgeräte. In: [Rot05c], S. 387–424

[Rot05b] *Kapitel* 8. Ortsbezug. In: [Rot05c], S. 267–306

[Rot05c] ROTH, Jörg: *Mobile Computing: Grundlagen, Technik, Konzepte*. 2., aktualisierte Auflage. dpunkt.verlag GmbH, 2005

[RT99] ROYER, Elizabeth M. ; TOH, Chai-Keong: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. In: *IEEE Personal Communications* 6 (1999), April, Nr. 2, S. 46–55. http://dx.doi.org/10.1109/98.760423. – DOI 10.1109/98.760423

[RZFK00] RESNICK, Paul ; ZECKHAUSER, Richard ; FRIEDMAN, Eric ; KUWABARA, Ko: Reputation Systems. In: *Communications of the ACM* 43 (2000), Dezember, Nr. 12, S. 45–48. http://dx.doi.org/10.1145/355112.355122. – DOI 10.1145/355112.355122

[SA02] STAJANO, Frank ; ANDERSON, Ross: The Resurrecting Duckling: Security Issues for Ubiquitous Computing. In: *IEEE Computer* 35 (2002), April, Nr. 4, S. 22–26. http://dx.doi.org/10.1109/MC.2002.1012427. – DOI 10.1109/MC.2002.1012427

[SBC+05] STERNE, D. ; BALASUBRAMANYAM, P. ; CARMAN, D. ; WILSON, B. ; TALPADE, R. ; KO, C. ; BALUPARI, R. ; TSENG, C.-Y. ; BOWEN, T. ; LEVITT, K. ; ROWE, J.: A General Cooperative Intrusion Detection Architecture for MANETs. In: *Proceedings of the 3rd IEEE International Workshop on Information Assurance (IWIA'05)*, 2005, S. 57–70

[Sch96] SCHNEIER, Bruce: *Applied Cryptography*. Wiley Computer Publishing, 1996

[Sch99] SCHNEIER, Bruce: Attack Trees – Modeling security threats. In: *Dr. Dobb's Journal* (1999), Dezember. http://www.schneier.com/paper-attacktrees-ddj-ft.html

[Sch00] SCHNEIER, Bruce: *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, Inc., 2000

[SDL+02] SANZGIRI, K. ; DAHILL, B. ; LEVINE, B. N. ; SHIELDS, C. ; BELDING-ROYER, E. M.: A Secure Routing Protocol for Ad hoc Networks. In: *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'02)*, 2002, S. 78–87

[SIR01] STUBBLEFIELD, Adam ; IOANNIDIS, John ; RUBIN, Aviel D.: Using the Fluhrer, Mantin, and Shamir Attack to Break WEP / AT&T Labs. Version: August 2001. http://citeseer.ist.psu.edu/stubblefield01using.html. 2001 (TD-4ZCPZZ). – Technical Report

[SIR04] STUBBLEFIELD, Adam ; IOANNIDIS, John ; RUBIN, Aviel D.: A Key Revovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP). In: *ACM Transactions on Information and System Security (TISSEC)* 7 (2004), Mai, Nr. 2, S. 319–332. http://dx.doi.org/10.1145/996943.996948. – DOI 10.1145/996943.996948

[SK03] SCHOLLMEIER, Rüdiger ; KUNZMANN, Gerald: GnuViz – Mapping the Gnutella Network to its Geographical Locations. In: *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 26 (2003), Nr. 2, S. 74–79

[SLO04] SCHMOYER, Tim R. ; LIM, Yu X. ; OWEN, Henry L.: Wireless Intrusion Detection and Response: A classic study using main-in-the-middle attack. In: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004)* Bd. 2, 2004, 883–888

[SM07] SUN MICROSYSTEMS, Inc.: *JXTA Java Standard Edition v2.5: Programmers Guide*. Version: September 2007. https://guest@jxta-guide.dev.java.net/svn/jxta-guide/trunk/src/guide_v2.5/JXSE_ProgGuide_v2.5.pdf, Abruf: August 2008

[SP05] SOMMER, Robin ; PAXSON, Vern: Exploiting Independent State For Network Intrusion Detection. In: *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC'05)*. Tucson, Arizona, USA, Dezember 2005

[SR96] STEELE, Guy L. (Hrsg.) ; RAYMOND, Eric S. (Hrsg.): *The New Hacker's Dictionary*. The MIT Press, 1996 http://www.catb.org/~esr/jargon/

[SS05] SABATER, Jordi ; SIERRA, Carles: Review on Computational Trust and Reputation Models. In: *Artificial Intelligence Review* 24 (2005), September, Nr. 1, S. 33–60. http://dx.doi.org/10.1007/s10462-004-0041-5. – DOI 10.1007/s10462–004–0041–5

[SSHW88] SEBRING, Michael M. ; SHELLHOUSE, E. ; HANNA, M. E. ; WHITEHURST, R. A.: Expert Systems in Intrusion Detection: A Case Study. In: *Proceedings of the 11th National Computer Security Conference*, 1988

[SST92] SNAPP, Steven R. ; SMAHA, Stephen E. ; TEAL, Daniel M.: The DIDS (Distributed Intrusion Detection System) Prototype. In: *Proceedings of the Summer 1992 USENIX Conference*. San Antonio, Texas, USA, Juni 1992, S. 227–234

[Ste08] STEINBRECHER, Sandra: *Mehrseitige Sicherheit in Reputationssystemen. Anforderungsanalyse, Gliederung, Umsetzungsmöglichkeiten*, Technische Universität Dresden, Diss., Juni 2008

[SW05] STEINMETZ, Ralf (Hrsg.) ; WEHRLE, Klaus (Hrsg.): *LNCS*. Bd. 3485/2005: *Peer-to-Peer Sytems and Applications*. Springer Verlag, 2005. http://dx.doi.org/10.1007/11530657. http://dx.doi.org/10.1007/11530657

[SWP03] SUN, Bo ; WU, Kui ; POOCH, Udo W.: Alert Aggregation in Mobile Ad Hoc Networks. In: *Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe '03)*. New York, NY, USA : ACM Press, 2003. – ISBN 1–58113–769–9, S. 69–78

[SZ00] SPAFFORD, Eugene H. ; ZAMBONI, Diego: Intrusion detection using autonomous agents. In: *Computer Networks* 34 (2000), Nr. 4, S. 547–570. http://dx.doi.org/10.1016/S1389-1286(00)00136-5. – DOI 10.1016/S1389–1286(00)00136–5. – ISSN 1389–1286

[TWP07] TEWS, Erik ; WEINMANN, Ralf-Philipp ; PYSHKIN, Andrei: Breaking 104 bit WEP in less than 60 seconds. In: *Cryptology ePrint Archive* Report 2007/120 (2007). http://eprint.iacr.org/2007/120

[VGM04] VLADIMIROV, Andrew A. ; GAVRILENKO, Konstantin V. ; MIKHAILOVSKY, Andrei A.: *Wi-Foo – The Secrets of Wireless Hacking*. Pearson Education, 2004

[VH02] VERWOERD, Theuns ; HUNT, Ray: Intrusion detection techniques and approaches. In: *Computer Communications* 25 (2002), 1356–1365. http://dx.doi.org/10.1016/S0140-3664(02)00037-3. – DOI 10.1016/S0140–3664(02)00037–3

[VHM05] VOSS, Marco ; HEINEMANN, Andreas ; MÜLHÄUSER, Max: A Privacy Preserving Reputation System for Mobile Information Dissemination Networks. In: *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm'05)*, 2005, S. 171–181

[Vil05] VILJANEN, Lea: Towards an Ontology of Trust. In: KATSIKAS, Sokratis (Hrsg.) ; LÓPEZ, Javier (Hrsg.) ; PERNUL, Günther (Hrsg.): *Trust, Privacy, and Security in Digital Business. Proceedings of the 2nd International Conference, TrustBus*. Copenhagen, Denmark : Springer Verlag, August 2005 (LNCS 3592), S. 175–184

[VIS] *Gesellschaft für Informatik, Fachgruppe 2.5.3: Verlässliche IT-Systeme*. http://www.iig.uni-freiburg.de/gi/vis/, Abruf: Juli 2008. – Web-Seite

[VK83] VOYDOCK, Victor L. ; KENT, Stephen T.: Security Mechanisms in High-Level Network Protocols. In: *ACM Computing Survey* 15 (1983), Juni, Nr. 2, S. 135–171. http://dx.doi.org/10.1145/356909.356913. – DOI 10.1145/356909.356913

[voi] *Void11 802.11b penetration testing utility*. http://www.wlsec.net/void11, Abruf: Mai 2008. – Projekt Web-Seite

[Vos04] VOSS, Marco: Privacy Preserving Online Reputation Systems. In: DESWARTE, Yves (Hrsg.) ; CUPPENS, Frédéric (Hrsg.) ; JAJODIA, Sushil (Hrsg.) ; WANG, Lingyu (Hrsg.): *Information Security Management, Education and Privacy, IFIP 18th World Computer Congress, TC11 19th International Information Security Workshops*. Toulouse, France : Kluwer, August 2004

[Wal00] WALKER, Jesse R.: Unsafe at Any Key Size; An Analysis of the WEP Encapsulation / Intel Corporation. 2000 (802.11-00/362). – IEEE Document

[WFP96] WHITE, Gregory B. ; FISCH, Eric A. ; POOCH, Udo W.: Cooperating Security Managers: A Peer-Based Intrusion Detection System. In: *IEEE Network* 10 (1996), Januar, Nr. 1, S. 20–23. http://dx.doi.org/10.1109/65.484228. – DOI 10.1109/65.484228

[WGR05] WEHRLE, Klaus ; GÖTZ, Stefan ; RIECHE, Simon: Distributed Hash Tables. Version: 2005. http://dx.doi.org/10.1007/11530657_7. In: [SW05], Kapitel 7, S. 79–93

[Whi04] WHITTEN, Alma: *Making Security Usable*. Pittsburgh, PA, Carnegie Mellon University, Diss., Mai 2004. – CMU-CS-04-135

[Wie04] WIEHLER, Gerhard: *Mobility, Security und Web Services*. Publicis Corporate Publishing, 2004

[WP00] WOLF, Gritta ; PFITZMANN, Andreas: Properties of protection goals and their integration into a user interface. In: *Computer Networks* 32 (2000), S. 685–699. http://dx.doi.org/10.1016/S1389-1286(00)00029-3. – DOI 10.1016/S1389–1286(00)00029–3

[Wri03] WRIGHT, Joshua: *Detecting Wireless LAN MAC Address Spoofing*. Online-Publikation. http://www.uninett.no/wlan/download/wlan-mac-spoof.pdf. Version: Januar 2003

[WT99] WHITTEN, Alma ; TYGAR, J.D.: Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In: *Proceedings of the 8th USENIX Security Symposium*. Washington, D.C., 1999

[WVE] *Wireless Vulnerabilities & Exploits*. http://www.wirelessve.org/, Abruf: Mai 2008. – Projekt Web-Seite

[WZS04] WINTER, Rolf ; ZAHN, Thomas ; SCHILLER, Jochen: DynaMO: A Topology-Aware P2P Overlay Network for Dynamic, Mobile Ad-Hoc Environments. In: *Telecommunication Systems* 27 (2004), Oktober, Nr. 2–4, S. 321–345. http://dx.doi.org/10.1023/B:TELS.0000041014.05554.ee. – DOI 10.1023/B:TELS.0000041014.05554.ee

[YLY+04] YANG, Hao ; LUO, Haiyun ; YE, Fan ; LU, Songwu ; ZHANG, Lixia: Security in Mobile Ad Hoc Networks: Challenges and Solutions. In: *IEEE Wireless Communications* 11 (2004), Februar, Nr. 1, S. 38–47. http://dx.doi.org/10.1109/MWC.2004.1269716. – DOI 10.1109/MWC.2004.1269716

[ZA02] ZAPATA, Manel G. ; ASOKAN, N.: Securing Ad hoc Routing Protocols. In: *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*. Atlanta, Georgia, USA, September 2002, S. 1–10

[Zah06] ZAHN, Thomas: *Structured Peer-to-Peer Services for Mobile Ad Hoc Networks*, Freie Universität Berlin, Diss., Juli 2006. http://www.diss.fu-berlin.de/2006/471/

[Zap02] ZAPATA, Manel G.: Secure Ad hoc On-Demand Distance Vector Routing. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 6 (2002), Juli, Nr. 3, S. 106–107. http://dx.doi.org/10.1145/581291.581312. – DOI 10.1145/581291.581312

[Zap06] ZAPATA, Manel G.: *Secure Ad hoc On-Demand Distance Vector (SAODV) Routing*. Internet Draft. http://www.potaroo.net/ietf/idref/draft-guerrero-manet-saodv/. Version: September 2006

[Zen89] ZENTRALSTELLE FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *IT-Sicherheitskriterien. Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik (IT)*. 1. Fassung vom 11.1.1989. Köln : Bundesanzeiger, 1989 http://www.bsi.bund.de/zertifiz/itkrit/itgruend.pdf

[ZH99] ZHOU, Lidong ; HAAS, Zygmunt J.: Securing Ad Hoc Networks. In: *IEEE Network* 13 (1999), November, Nr. 6, S. 24–30. http://dx.doi.org/10.1109/65.806983. – DOI 10.1109/65.806983

[ZL00] ZHANG, Yongguang ; LEE, Wenke: Intrusion Detection in Wireless Ad-Hoc Networks. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 2000

[ZLH03] ZHANG, Yongguang ; LEE, Wenke ; HUANG, Yi-An: Intrusion Detection Techniques for Mobile Wireless Networks. In: *Wireless Networks* 9 (2003), September, Nr. 5, S. 545–556. http://dx.doi.org/10.1023/A:1024600519144. – DOI 10.1023/A:1024600519144

[ZY02] ZEINALIPOUR-YAZTI, Demetrios: Exploiting the Security Weaknesses of the Gnutella Protocol / Department of Computer Science, University of California. Version: März 2002. http://www.cs.ucr.edu/~csyiazti/courses/cs260-2/project/gnutella.pdf. 2002. – Course Project for Seminar in Computer Security